



UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Transformando dados em conhecimento: LADDRES, uma aplicação prática

Trabalho de Conclusão de Curso

Nélson Rangel Santos Passos



São Cristóvão – Sergipe

2018

UNIVERSIDADE FEDERAL DE SERGIPE
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

Nélson Rangel Santos Passos

**Transformando dados em conhecimento: LADDRES, uma
aplicação prática**

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador(a): Hendrik T. Macedo
Coorientador(a): Bruno O. P. Prado

São Cristóvão – Sergipe

2018

*Dedico esta conquista a todos aqueles que
compartilharam e fizeram parte desta jornada.*

*Família, amigos e professores
que doaram parte de si para que tudo isso fosse possível.*

Agradecimentos

Depois de uma longa jornada de 9 anos de vida universitária, que incluíram troca de curso, greves, dedicação a pesquisa, graduação sanduíche, mais greves, dedicação a carreira profissional, festas e muitas horas de estudo, resumir tudo que há para ser agradecido durante todo esse tempo em uma só página é um desafio.

Nada mais justo do que começar agradecendo àqueles que foram os principais responsáveis pela minha formação pessoal e que com todo seu amor e dedicação me tornaram o que eu sou hoje: minha família. Aos meus pais, por serem um poço de amor e carinho, espelhos de como eu devo e quero ser. Amo vocês! A minha irmã, por quem eu sinto o mais puro e verdadeiro amor. Ao último integrante da família, meu companheiro de TCC nas madrugadas, Arroz, “o neném mais lindo do mundo” (Passos, 2018). Agradecer, também, aqueles que me amam desde antes mesmo de eu ter consciência de existir e aos que construíram este sentimento durante a vida: avós, avô, madrinha, padrinho, tias, tios, primas e primos, muito obrigado!

Agradecer aos meus amigos com quem eu tive a honra de compartilhar grandes momentos da vida. Aos Amigos Lindos, que estão presentes desde sempre, juntos há mais de uma década, Alan, Alyne, Bia, Danilo, Jefferson, João Paulo, Lucas, Rafael e Zé, da Escolinha Dengoso pra vida! Aos que surgiram durante a vida universitária, o parceiro de engenharias e “sócio” Matheus Iácono, aos engenheiros Jefferson e Lucas, ao parceiro de facas e cozinha Miguel Felipe, a galera do dominó e do Carnaueira, aos amigos do Ciência sem Fronteiras, aos amigos da Atos Capital e a todos aqueles que deixaram comigo um pedacinho de si durante essa jornada.

Um agradecimento especial a minha namorada Camila, que com todo seu amor, companheirismo, incentivo e atitudes, foi peça fundamental na conclusão deste ciclo e parte importante em minha vida desde que passou a fazer parte dela <3.

Aos professores, por sua dedicação na arte que é lecionar/orientar tantos jovens durante os anos mais intensos de suas vidas. Um agradecimento especial a professora Maria Augusta, por me mostrar o caminho da ciência e por, em 12 meses de pesquisa, ter nos conduzido na produção de artigos, softwares e patentes. Aos professores Bruno e Hendrik, que são brilhantes em sala de aula e que, juntos com o grupo Ludiico, nos ajudaram na condução e produção do projeto descrito neste documento.

Por último, mas não menos importante, agradecer ao meu colega de curso e companheiro no projeto LADDRES, Ariel, por sua dedicação neste e em todos os trabalhos que fizemos juntos: aplicativos, artigos, apresentações todas as sextas-feiras... Foi um prazer trabalhar com você!

Obrigado, Deus, por permitir que todas essas pessoas maravilhosas pudessem fazer parte do meu caminho e que eu pudesse viver experiências tão ricas ao longo da vida!

A vida é tipo roda-gigante

Então pra que esculacho?

Hoje você tá em cima

Amanhã tá embaixo

[...]

A humildade é a essência da vida

(Black Style)

Resumo

Democracia é um regime político baseado na escolha da maioria. Porém, uma escolha consciente só é possível quando se tem acesso a informação de qualidade. Este trabalho teve como objetivo agregar dados de diferentes fontes e transformá-los em informação para os eleitores brasileiros. Através de métodos de ETL (Extract, Transform, Load) aplicados a dados abertos e dados proprietários, este trabalho construiu um processo que cobre as etapas de obtenção e transformação de dados, geração de datasets, modelagem e população de banco de dados, desenvolvimento de uma API REST para acesso público às informações e construção de um aplicativo para dispositivos móveis como modelo de visualização do conhecimento gerado. Ao todo, para as eleições de 2018, foram processados quase dois milhões de candidaturas, meio milhão de atuações parlamentares e cinco mil processos judiciais. Além disso, a API REST disponibilizada ao público registrou pouco mais de três milhões de acessos e o aplicativo móvel superou a marca de doze mil downloads durante a última semana de corrida eleitoral para o primeiro turno.

Palavras-chave: Dados Abertos, Mineração de Dados, ETL, Eleições.

Abstract

Democracy is a political regime based on the majority's choice. However, people can only make conscious decisions if they have access to high quality information. This paper aimed to join data from different sources and to transform them in knowledge to brazilians voters. It applied ETL (Extract, Transform, Load) methods on open and property data to build a proccess that covers data gathering and transformation, dataset generation, database modeling and population, public APIs development, and a mobile app as the knowledge's visualization model. As a result, for 2018 Brazil's general elections, it processed almost two million candidacies, half a million deputies' tasks and five thousand court lawsuits. Furthermore, the products released by this research reached good performance indicators: the access logs recorded more than three million hits for the public API and twelve thousand downloads for the mobile app in the last week of the first-round's political campaign.

Keywords: Open Data, Data Mining, Development Proccess, Elections.

Lista de ilustrações

Figura 1 – Captura de tela do andamento do projeto no aplicativo Trello	25
Figura 2 – Fluxo de desenvolvimento do projeto.	26
Figura 3 – Lista com todos os candidatos, agrupados por cargo.	29
Figura 4 – Cartão com o perfil do candidato.	30
Figura 5 – Resumo da vida pública do candidato.	30
Figura 6 – Possíveis classificações para o posicionamento do candidato.	31
Figura 7 – Descrição do tema, posicionamento referenciado e fontes de informação. . .	31
Figura 8 – Posicionamento do candidato sobre temas que ganharam relevância.	31
Figura 9 – Lista de processos nos quais o candidato é réu.	32
Figura 10 – Candidato não responde a nenhum processo judicial.	32
Figura 11 – Detalhes de candidaturas anteriores.	32
Figura 12 – Detalhes de atuação em mandatos anteriores.	32
Figura 13 – Organograma do Poder Judiciário	39
Figura 14 – Estrutura do Banco de Dados do Projeto	45
Figura 15 – Estrutura das <i>hot tables</i> geradas.	46
Figura 16 – Página de captura do projeto	52
Figura 17 – Solicitação de acesso à localização do usuário.	55
Figura 18 – Respostas a perguntas frequentes de usuários.	55
Figura 19 – Seletor para indicar a localização do usuário.	55
Figura 20 – Tela principal, exibe todos os candidatos desta eleição.	55
Figura 21 – Tela de seleção de filtros para exibição dos candidatos.	55
Figura 22 – Cartão com o perfil do candidato.	56
Figura 23 – Cartões com resumo e processos de um candidato.	56
Figura 24 – Cartão com os processos judiciais de um candidato.	56
Figura 25 – Cartão com o histórico de candidaturas de um candidato.	56

Lista de códigos

Código 1	– Script para <i>download</i> dos <i>datasets</i> do TSE	34
Código 2	– Retorno simplificado de um Deputado Federal	38
Código 3	– Retorno detalhado para um Deputado Federal	38
Código 4	– Parte do arquivo disponibilizado pelo projeto Vigie Aqui	41
Código 5	– Migração de dados de um Pandas Dataframe para o banco de dados MySQL	44
Código 6	– Exemplo de algoritmo para geração de uma <i>hot table</i>	47
Código 7	– Configuração do Nginx para servir a API do LADDRES	49
Código 8	– Retorno de uma consulta à API de Candidatos	54

Sumário

1	Introdução	11
2	Referencial Teórico	15
2.1	Dados Abertos	15
2.1.1	Dados Conectados (Linked Data)	16
2.1.2	Governo Aberto	16
2.2	Engenharia de Software	18
2.2.1	Modelos de Desenvolvimento	18
2.2.2	Estruturas de Armazenamento de Dados	19
2.2.3	APIs	20
2.2.4	Aplicativos Móveis	21
2.2.5	Páginas de Captura	22
3	Materiais e Métodos	24
3.1	Definição do Escopo	26
3.2	Prototipação do App	27
3.3	Obtenção e Tratamento de Dados	27
3.4	Banco de Dados	28
3.5	API	28
3.6	Aplicativo para <i>Smartphones</i>	28
4	Desenvolvimento	29
4.1	Prototipação do App	29
4.2	Obtenção e Tratamento de Dados	31
4.2.1	Tribunal Superior Eleitoral	32
4.2.2	Câmara de Deputados	36
4.2.3	Processos Judiciais	39
4.2.4	Dados Não Estruturados	42
4.3	Banco de Dados	43
4.4	API	48
4.5	Aplicativo para <i>Smartphones</i>	50
4.6	Página de Captura	51
5	Resultados e Discussões	53
5.1	Qualitativo	53
5.1.1	API Pública	53

5.1.2	Aplicativo Móvel	54
5.2	Quantitativo	57
6	Considerações Finais	58
	Referências	60

1

Introdução

O estado democrático brasileiro é bastante frágil. Em 126 anos de República, poucos foram os comandantes que conseguiram cumprir o que determina a constituição e completar os seus ciclos de governo. Seja derrubado por golpe ou por processos de impeachment, após o governo de Juscelino Kubitschek (1955-1961), dos presidentes eleitos democraticamente, apenas Fernando Henrique Cardoso e Luiz Inácio Lula da Silva permaneceram no cargo mais importante do país até o fim de todos os seus mandatos ([JOKURA, 2016](#)).

Em 2013, vivemos um momento que ficou marcado na história do país e que desencadeou uma série de eventos importantes para compreender o atual cenário político brasileiro. Uma onda de protestos mobilizou milhares de pessoas em diversas cidades brasileiras: as ruas foram ocupadas, em catarse, por diversos segmentos sociais, num movimento, inicialmente apartidário, que ficou conhecido como “Jornadas de Junho”. O ano de 2013 representou um grande grito de insatisfação com a representação política, encorajando outros movimentos da sociedade civil, com diferentes pautas e interesses, a se manifestar nos anos seguintes ([CHARLEAUX, 2017](#)).

Após uma combinação de escândalos de corrupção durante as obras da copa do mundo de 2014 ([LEITE, 2017](#)) e dos jogos olímpicos de 2016 ([BOUCHEZ; GATINOIS, 2017](#)), eleições conturbadas que polarizaram o país e erros em políticas públicas que reduziram a capacidade de crescimento da economia brasileira ([FILHO, 2017](#)), o Brasil entrou em uma das maiores crises econômicas de sua história recente. Somado a isso, tivemos o avanço da Operação Lava Jato, que acabou revelando relações escusas entre políticos tradicionais brasileiros e empresas privadas, entre elas as maiores construtoras do país, que se beneficiaram de esquemas fraudulentos e de superfaturamento de obras públicas durante mais de vinte anos de governos ([FOLHA DE SÃO PAULO, 2018](#)).

Contrariando a previsão do deputado federal, eleito pelo estado de São Paulo, Francisco Everardo Oliveira Silva - o Tiririca -, cuja campanha baseou-se no slogan “pior do que tá não fica” ([YOUTUBE, 2010](#)), em 17 de abril de 2016, o Brasil conheceu, em horário nobre da televisão

brasileira, o total despreparo de grande parte de seus deputados:

Pela paz em Jerusalém, pelos maçons do Brasil, pela nação evangélica, pelos militares de 64. Pelos pais, pelos filhos e pelos netos, os que já existem e os que “estão chegando”. A votação do processo de impeachment de Dilma Rousseff durou mais de cinco horas e teve de tudo: gritos de apoio, cuspidelas, insultos e justificações um tanto ou quanto fora do guião. No final, o “tchau, querida” acabaria por vencer ao “não vai ter golpe” com 367 votos a favor da destituição de Dilma Rousseff e os parlamentares festejaram como se o Brasil tivesse vencido a “Copa” (CORREIA, 2016).

Meses depois da votação do impeachment, boa parte dos representantes democraticamente eleitos pela população brasileira estava intimamente ligada a escândalos de corrupção, incluindo o atual presidente, líderes do governo e líderes da oposição (SARDINHA, 2016).

Em um cenário político tão caótico quanto este, a sociedade civil anseia por uma mudança radical. Se por um lado o Brasil encontra-se entre os países mais corruptos do mundo (G1, 2018), ele aparece, também, no topo da lista de países mais transparentes do globo (OPEN KNOWLEDGE INTERNATIONAL, 2016). Iniciativas como o Portal da Transparência (do poder executivo federal) e os programas de dados abertos do poder legislativo, principalmente da câmara de deputados federal, trazem uma boa oportunidade para se contornar este cenário. A sociedade tem, pela primeira vez, um poderoso conjunto de ferramentas tecnológicas que podem ajudar a iniciar um novo capítulo na história política do país.

Dados abertos, especialmente os governamentais, são um ótimo recurso ainda muito pouco explorado (GRAY et al., 2011). Apesar disso, a partir dessa política de abertura de dados do governo brasileiro, a sociedade civil começou a apresentar ferramentas numa tentativa de empoderar os cidadãos e denunciar o não cumprimento das leis, ou, em muitos casos, a imoralidade delas. Entretanto, muitas destas iniciativas acabam sendo descontinuadas por falta de verbas ou por falta de aderência coletiva.

Um dos melhores exemplos em termos de estruturação de projeto e engajamento social é a Operação Serenata de Amor, um projeto de código aberto que utiliza ciência de dados e inteligência artificial para controle social da administração pública (DATA SCIENCE BRIGADE, 2018). A Serenata de Amor utiliza dados abertos da câmara dos deputados para identificar irregularidades na utilização da cota parlamentar e tem alcançado resultados expressivos dentro¹ e fora² do Brasil. O projeto foi iniciado a partir de um financiamento coletivo que arrecadou R\$ 80.000,00 durante a primeira campanha e que hoje é mantido por mais de 600 apoiadores de todas as partes do mundo (APOIA.SE, 2018).

¹ <<http://g1.globo.com/fantastico/noticia/2017/11/conheca-brasileiros-que-resolveram-denunciar-corrupcao-de-governantes.html>>

² <<https://changelog.com/podcast/268>>

Na área de acompanhamento das atividades dos agentes públicos, projetos como o Quem Me Representa³, o Eu Parlamentar⁴, o Match Eleitoral⁵ e o #TemMeuVoto⁶ tem como objetivo testar a similaridade ideológica entre os parlamentares e o usuário; baseado nas respostas do eleitor a questões polêmicas votadas na câmara de deputados durante a última legislatura e/ou amplamente discutidas pela mídia nos últimos anos, os aplicativos retornam uma lista com os parlamentares cujos votos estão mais alinhados ao posicionamento do usuário.

Uma outra abordagem que tem ganhado destaque é a do ranqueamento dos parlamentares, que classifica senadores e deputados federais do melhor para o pior, baseado em critérios próprios de cada projeto. Entre os maiores destaques estão o Atlas Político⁷ e o Ranking dos Políticos⁸.

Outro projeto que utiliza dados abertos do governo federal, só que numa outra vertente, é o QEDu⁹, uma iniciativa que tem como missão dar vida às estatísticas a fim de promover melhores escolhas na educação (QEDU, 2018). O projeto mapeou a situação da educação básica brasileira, através do cruzamento de dados oficiais de diferentes fontes, e tem buscado criar conexões para entender a situação e produzir melhorias significativas na qualidade da educação pública no país.

Apesar do surgimento de um número cada vez maior de projetos que consomem dados abertos do poder público, poucos deles apresentam riqueza de detalhes, transparência e imparcialidade simultaneamente.

Com a iminência das eleições federais e estaduais de 2018, e por acreditar que a qualidade da informação que chega ao eleitor define a qualidade da democracia de um país, o objetivo deste trabalho é utilizar dados abertos do governo brasileiro em conjunto com outras fontes de dados estruturados e não-estruturados para proporcionar a possibilidade de que qualquer brasileiro tenha informação transparente - cujas fontes originais estão disponíveis para consulta -, imparcial - sem defender nenhum posicionamento político - e de qualidade - que responda aquilo que o eleitor quer saber - sobre todos os candidatos aos cargos eletivos deste ano.

Com seu nome e modelo inicial inspirados pela Plataforma Lattes - plataforma virtual que integra as bases de dados de currículos, grupos de pesquisa e instituições que produzem Ciência e Tecnologia no Brasil (MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO, 2018) - o Laboratório de Análise de Desempenho Dos REpresentantes da Sociedade (LADDRES) tem como objetivo unificar e integrar as biografias de todos os agentes públicos que participaram da história recente da democracia no país. Através do LADDRES, eleitores de qualquer idade e grau de instrução podem, não apenas conhecer a trajetória política de cada candidato, como

³ <<http://qmrepresenta.com.br/>>

⁴ <<https://play.google.com/store/apps/details?id=com.ionicframework.eu273435>>

⁵ <<https://matcheleitoral.folha.uol.com.br/>>

⁶ <<https://temmeuvoto.com/>>

⁷ <<http://www.atlaspolitico.com.br/>>

⁸ <<http://www.politicos.org.br/>>

⁹ <<http://www.qedu.org.br/>>

conhecer seu posicionamento perante temas amplamente discutidos pela mídia e pela sociedade, além dos detalhes de sua atuação parlamentar durante cada um de seus mandatos.

Para alcançar este objetivo geral, o trabalho foi dividido em objetivos específicos que são:

1. Desenvolver uma estrutura de mineração e extração de dados abertos a partir de diversas fontes e formatos;
2. Tratar e padronizar as informações que agregarão valor à solução final (montar um currículo);
3. Modelar e povoar uma estrutura de banco de dados que servirá como base de busca de informações por uma interface de acesso pública;
4. Construir uma API para disponibilização dos dados tratados e modelados para o público;
5. Desenvolver um aplicativo para dispositivos móveis que possuem sistema operacional Android e iOS.
6. Por fim, como ferramenta de *marketing* e divulgação do projeto, construir uma página de captura e aquisição de usuários para o aplicativo (*landing page*) e divulgá-la em redes sociais para todo Brasil.

O documento está dividido em seis capítulos. Após este capítulo introdutório, são apresentados, no Capítulo 2, os referenciais teóricos necessários para o desenvolvimento do projeto. No Capítulo 3 são expostos os materiais e métodos utilizados durante o trabalho. Já o Capítulo 4 detalha, passo-a-passo, o desenvolvimento de cada uma das etapas com foco na aplicação prática do LADDRES. No 5º Capítulo são mostrados os resultados obtidos, quantitativa e qualitativamente. Por fim, no Capítulo 6, o trabalho é concluído com considerações finais sobre o processo desenvolvido e sugestões para trabalhos futuros.

2

Referencial Teórico

O referencial teórico se concentra em duas grandes áreas da Computação: (1) Dados Abertos, para o processo de obtenção de dados e geração de conjuntos de dados (*datasets*), e (2) Engenharia de Software, para o desenvolvimento de toda infraestrutura de banco de dados, *APIs* (*Application Programming Interfaces*) e aplicações para o usuário final.

2.1 Dados Abertos

Apesar de o movimento de dados abertos ter ganhado destaque em escala global somente na última década, o termo “Dados Abertos” surgiu em 1995 em um documento elaborado por uma agência científica norte-americana, abordando a divulgação de dados ambientais e geofísicos ([CHIGNARD, 2013](#)). Durante a produção do documento, os autores promoveram um completo e aberto intercâmbio de informação científica entre os diversos laboratórios espalhados pelo mundo, a fim de alcançar as condições necessárias para análise e compreensão dos fenômenos naturais de maneira global.

O movimento alcançou proporções internacionais quando Barack Obama, então presidente dos Estados Unidos, redefiniu o conceito de Governo Eletrônico, ao determinar que seu governo deveria atender aos princípios de transparência, participação e colaboração ([OBAMA, 2009](#)). O governo norte-americano, então, reelaborou sua estrutura, incorporando em definitivo o conceito de Dados Abertos e Dados Abertos Conectados ([RIBEIRO; ALMEIDA, 2013](#)).

Segundo a Open Definition, um projeto da Open Knowledge International, dados abertos são dados que podem ser livremente acessados, utilizados, modificados e compartilhados por qualquer pessoa, com qualquer propósito - sujeito, no máximo, à exigência de atribuição da fonte dos dados e de compartilhá-los sob a mesma licença em que foram obtidos ([OPEN DEFINITION, 2018](#)).

Segundo a Open Knowledge International, o conceito de dados abertos está sustentando

sobre três pilares ([OPEN KNOWLEDGE INTERNATIONAL, 2018](#)):

1. Disponibilidade e acesso: os dados devem estar disponibilizados como um todo e sob um custo não maior que um custo razoável de reprodução. Além disso, devem estar, preferencialmente, disponíveis para *download* pela internet. Os dados também devem estar disponíveis de uma forma conveniente e modificável;
2. Reutilização e redistribuição: os dados devem ser disponibilizados sob termos que permitam sua reutilização e sua redistribuição, inclusive a combinação com outros conjuntos de dados. Além disso, os dados devem ser processáveis por máquina;
3. Participação universal: qualquer pessoa deve ser capaz de usar, reutilizar e redistribuir; não deve haver discriminação contra áreas de atuação ou contra pessoas ou grupos. Por exemplo, restrições de uso “não comerciais” que poderiam impedir uso “comercial” ou restrições de uso para certos fins, como apenas educacionais, não são permitidos.

2.1.1 Dados Conectados (Linked Data)

Apesar de termos um volume cada vez maior de dados abertos disponibilizados por entidades públicas e privadas, nem todos os conjuntos de dados estão bem estruturados e padronizados, disponibilizados em formato de fácil processamento e não proprietário, ou apresentados sob a ótica de contextualização.

A fim de criar um modelo moderno, de fácil processamento e acesso global, surgiu o conceito de Dados Conectados. Fundamentadas em tecnologias web, como HTTP e URI, com o objetivo de permitir a leitura de dados conectados, de forma automática, por agentes de software, o conceito de Dados Conectados (Linked Data) pode ser definido como um conjunto de boas práticas de publicação e conexão de conjuntos de dados. A partir dessa rede de conjunto de dados e da integração semântica dos próprios dados, inúmeras novas aplicações podem ser desenvolvidas ([AGUNE; FILHO; BOLLIGER, 2010](#)), apresentando uma mudança na abordagem da estruturação da web: saindo de uma estrutura que funciona apenas por meio de links e conexões entre documentos, para uma estrutura onde documentos e dados também passam a ser ligados, em que deve ser uma prática expor e compartilhar os elementos de ligação de dados, tanto às informações, como aos conhecimentos ([RIBEIRO; ALMEIDA, 2013](#)).

2.1.2 Governo Aberto

Um outro conceito que merece ainda mais destaque neste trabalho é o conceito de dados abertos governamentais ou governo aberto. O termo serve para denominar a “disponibilização, através da Internet, de informações e dados governamentais de domínio público para a livre utilização pela sociedade” ([AGUNE; FILHO; BOLLIGER, 2010](#)). Este conceito remete à defesa dos dados públicos como pertencentes aos cidadãos, que deveriam ter acesso irrestrito às

informações governamentais (VAZ; RIBEIRO; MATHEUS, 2010). A capacidade de compartilhar, tratar e reutilizar estas informações é fator determinante para acompanhar as ações do governo e pressionar as lideranças políticas por uma melhor gestão e controle dos gastos públicos.

Em dezembro de 2007, um grupo de trinta especialistas, defensores do conceito de governo aberto, se reuniram na Califórnia e definiram os oito princípios para dados abertos governamentais (OPEN GOVERNMENT DATA, 2018). Segundo eles, para serem considerados abertos, os dados do governo devem ser disponibilizados publicamente seguindo os princípios de serem:

1. Completos: todos os dados públicos estão disponíveis. Dados públicos são dados que não estão sujeitos a limitações válidas de privacidade, segurança ou controle de acesso.
2. Primários: os dados são apresentados assim como foram coletados da fonte, com o maior nível de granularidade possível, sem agregação ou modificações.
3. Atuais: os dados devem ser disponibilizados tão rápido quanto necessário à preservação de seu valor.
4. Acessíveis: os dados são disponibilizados para o maior alcance possível de usuários e o maior número possível de finalidades. Dados não são acessíveis se eles só puderem ser acessados por meio de formulários ou se o acesso por meios automatizados forem proibidos.
5. Processáveis por máquina: os dados são razoavelmente estruturados de modo que permitem processamento automatizado.
6. Não discriminatórios: os dados são disponibilizados a qualquer pessoa, sem exigência de registro prévio.
7. Não proprietários: os dados são disponibilizados num formato sobre o qual nenhuma entidade detenha controle exclusivo.
8. Livres de licenças: Os dados não estão sujeitos a nenhum direito autoral, patente, marca registrada ou segredo industrial. Restrições sensatas sobre privacidade, segurança e privilégios de acesso podem ser permitidas.

Os benefícios da adoção de dados abertos governamentais nos campos da transparência e do controle social são incontáveis. Além de contribuir para o aumento da transparência das ações do governo, permitindo um maior controle por parte da sociedade sobre o trabalho das lideranças políticas, há, também, a possibilidade de criação de novas informações e aplicações a partir dos dados governamentais, originando novos serviços e soluções para problemas do cotidiano da população (VAZ; RIBEIRO; MATHEUS, 2010). Para Diniz (2010): “A disponibilização

de dados governamentais abertos permite que as informações sejam utilizadas da maneira e conveniência do interessado de tal forma que elas possam ser misturadas e combinadas para agregar mais valor aos dados”.

O governo brasileiro foi um dos pioneiros na disponibilização de dados abertos no mundo, participando, inclusive, da fundação da Open Government Partnership, entidade criada em 2011 e que atualmente conta com a participação de 79 países ([OPEN GOVERNMENT PARTNERSHIP, 2018](#)). Com a aprovação da Lei de Acesso à Informação, que destina-se a assegurar o direito fundamental de acesso à informação ao cidadão brasileiro ([BRASIL, 2011](#)), aprovada em novembro de 2011, o governo brasileiro criou, também, o [dados.gov.br](#), portal que disponibiliza dados governamentais seguindo os princípios de dados abertos. Os dados abertos governamentais fazem parte da política de acesso à informação do governo federal (INDA), criado em 2012 ([ISOTANI; BITTENCOURT, 2015](#)), que vem se consolidando e conquistando espaço entre cientistas de dados e iniciativas da sociedade civil.

Além de institucionalizar a prática de dados abertos na esfera federal, o governo brasileiro tem protagonizado campanhas de fomento ao desenvolvimento de aplicações que utilizem esses conjuntos de dados. Um exemplo disso são os eventos promovidos ou apoiados por órgãos públicos tais como concursos, *hackathons*, *datathons*, *meetups* e expedições de dados que acabam gerando resultados como a criação de produtos, aplicativos, infográficos, visualizações etc ([PORTAL BRASILEIRO DE DADOS ABERTOS, 2018](#)).

2.2 Engenharia de Software

Em Engenharia de Software todo projeto começa com uma necessidade de negócio, uma visão da solução e o levantamento das restrições de custo e tempo. O primeiro passo é entender e delimitar o contexto de negócio em que o projeto está inserido ([MARTINS, 2007](#)). Após o levantamento dos requisitos e das necessidades de clientes e usuários, vem a elaboração de um projeto técnico, onde, dentre outras atividades, são definidos a metodologia e o fluxo de desenvolvimento, a arquitetura, as estruturas de armazenamento de dados e os padrões de projeto a serem aplicados.

2.2.1 Modelos de Desenvolvimento

O processo de desenvolvimento de um software é altamente imprevisível; seja pela falta de clareza que os idealizadores tem sobre o resultado final ou seja pelo fato de o usuário final ter expectativas diferentes da projetada no início do projeto. Os modelos de desenvolvimento de *software* tradicionais são duramente criticados pelo fato de seguirem uma linha não-incremental, inflexível e pouco aberta a mudanças ocorridas no decorrer desse complexo processo. As metodologias ágeis surgiram, então, como uma alternativa que se propunha a mitigar os problemas decorrentes dessas características dos modelos anteriores ([TANNER; WILLINGH et al., 2014](#)).

Resumidamente, o manifesto para o desenvolvimento ágil de software destaca e valoriza os seguintes pontos (BECK et al., 2001):

- Indivíduos e interações mais que processos e ferramentas;
- Software em funcionamento mais que documentação abrangente;
- Colaboração com o cliente mais que negociação de contratos;
- Responder a mudanças mais que seguir um plano.

Com um total de doze itens, o manifesto inspirou o surgimento de diversas metodologias de desenvolvimento, entre as mais adotadas podem ser destacadas (MIERS, 2017):

- Scrum: torna o desenvolvimento do sistema algo empírico e incremental, facilitando a absorção de mudanças no decorrer do projeto (GENARI; FERRARI, 2016);
- XP: voltado para projetos cujos requisitos são vagos e mudam com frequência, prioriza a agilidade no processo de desenvolvimento do projeto e o envolvimento do cliente em níveis extremos (NUNES, 2017);
- Kanban: desacopla as fases do processo de forma que cada uma dessas atividades possa ter sua própria cadência para melhor se ajustar à realidade e à necessidade que o processo demanda (GOMES, 2014).

Embora tenham abordagens um pouco diferentes, essas metodologias podem ser combinadas e aplicadas em diferentes momentos durante o amadurecimento de uma equipe. O combo Scrum + Kanban tem ganhado muito destaque na última década e tem sido uma alternativa mais flexível a ambas abordagens separadas (FRANCINO, 2018).

2.2.2 Estruturas de Armazenamento de Dados

O conceito de banco de dados pode ser sucintamente definido como uma coleção de dados organizada de maneira que permita a manipulação de grande volume de informação por meio de consultas e inserções (BIRGEN; PREISIG; MORUD, 2014). Seus modelos podem ser divididos, basicamente, entre bancos de dados relacionais e não-relacionais.

Os registros em um banco de dados relacional são organizados em esquemas bem definidos, agrupados por tabelas com linhas e colunas, em que todas as linhas têm o mesmo número de colunas. Nesta abordagem, o resultado mais comum é um projeto com múltiplas tabelas interligadas por chaves estrangeiras e que geralmente combina informações de mais de uma tabela para entregar o valor esperado. Quanto maior o número de tabelas e de registros,

maior o tempo despendido pelo banco de dados para retornar uma resposta ([ABOUTORABI et al., 2015](#)).

Já no modelo não relacional, não há necessidade de pré-determinar a estrutura que armazenará os dados. Outro ponto que ganha destaque é o fato das propriedades ACID (Atomicidade, Consistência, Isolamento e Durabilidade) de transações serem ignoradas ou pouco relevantes nesse modelo, acarretando num aumento de performance. Por outro lado, a falta de um esquema bem definido para seus elementos pode resultar em problemas para casos de consultas que envolvam mais de uma estrutura ou para estruturas que apresentem inconsistências entre suas "linhas" ([BUSSLER, 2015](#)).

Entre os principais bancos de dados relacionais, merecem destaque o Oracle, o MySQL, o SQL Server, o MariaDB e o PostgreSQL. Já entre os bancos de dados não relacionais, o MongoDB e o Redis são os maiores destaques ([DB-ENGINES, 2018](#)).

2.2.3 APIs

Para o desenvolvimento de uma Interface de Programação de Aplicativos (API) através da internet, é preciso definir alguns pontos como:

- Servidor HTTP;
- Linguagem de programação utilizada no desenvolvimento do *web service*;
- Arquitetura da interface de ligação entre servidor e aplicação.

Os dois servidores de código aberto mais populares do mundo são responsáveis por mais da metade do tráfego de dados da internet ([ELLINGWOOD, 2015](#)). Apesar de Apache HTTP Server e Nginx compartilharem muitas qualidades, as duas ferramentas apresentam soluções diferentes para a tarefa de servir páginas, arquivos de mídia e serviços online. O projeto de desenvolvimento do Apache foi o resultado de um esforço para construir e manter um servidor HTTP de código aberto que fosse robusto, eficiente e extensível para sistemas operacionais modernos, como Unix e Windows ([THE APACHE SOFTWARE FOUNDATION, 2018](#)). Com o aumento exponencial no volume de requisições que os servidores web tem que gerenciar paralelamente, surgiu a necessidade de uma nova solução que priorizasse sua capacidade de escala. Assim surgiu o Nginx. Criado pelo russo Igor Sysoev, o servidor foi desenvolvido para o concurso C10K Problem (desafio para criação de servidores que conseguissem gerenciar 10 mil conexões simultaneamente) e hoje é um dos servidores mais utilizados no mundo ([CISNEIROS; RAMOS, 2015](#)).

Escolher uma linguagem de programação para um projeto é sempre um desafio. Atualmente existem inúmeras linguagens e incontáveis *frameworks* disponíveis para desenvolvimento de aplicações *backend* (que rodam no servidor). A escolha depende de fatores que vão desde a

preferência da equipe até o tipo específico de problema que se deseja solucionar. Uma análise feita por Crawford e Hussain (2017) comparou itens como curva de aprendizagem, documentação e suporte, popularidade, qualidade das bibliotecas e desempenho em testes com PHP, Django (Python), Ruby on Rails (Ruby) e Node.js (Javascript). Sua conclusão foi a de que apesar de ainda não ter uma comunidade de desenvolvedores com o mesmo volume de integrantes e, assim, não ter o melhor suporte dentre os comparados, o Node.js mostrou-se, na média, como uma opção que merece bastante atenção de quem deseja iniciar um novo projeto web.

Em termos de arquitetura de interface entre aplicação e *web service*, as principais são a SOAP (Simple Object Access Protocol) e a REST (Representational State Transfer) (CASTILLO et al., 2011). Embora a arquitetura SOAP apresente vantagens como o gerenciamento de computação distribuída, a independência de protocolos de transporte e o suporte a outros padrões que não o web, sua verbosidade, dependência de ferramentas e longa curva de aprendizado abriram espaço para a aderência do padrão REST, que foi criado justamente para contornar esses problemas (MUMBAIKAR; PADIYA et al., 2013). Para alcançar o ganho de performance proposto, a arquitetura REST abriu mão de algumas abstrações (citadas como vantagens da arquitetura SOAP) e foi criada voltada para os padrões HTTP/web (CASTILLO et al., 2011).

2.2.4 Aplicativos Móveis

O aumento expressivo do número de usuário e de aplicações para dispositivos móveis tem tornado esta área uma das mais promissoras em engenharia de software (VAUPEL et al., 2018). Só no Brasil o número de celulares ativos ultrapassou o número de habitantes do país e, em abril de 2018, atingiu a marca de 220 milhões de dispositivos (DEMARTINI, 2018), contra aproximadamente 207 milhões de habitantes.

Atualmente é possível encontrar um aplicativo móvel para suprir qualquer necessidade. Seja para alugar uma bicicleta compartilhada, para realizar transações bancárias, para ajudar pessoas com deficiência visual a identificar objetos em casa¹ ou para doar água, comida e medicamentos para comunidades pobres sem gastar nenhum centavo², existe um aplicativo disponível como solução.

Tradicionalmente os aplicativos para *smartphone* eram desenvolvidos especificamente para um determinado sistema operacional (plataforma). Esta abordagem traz benefícios para o usuário final, por apresentar uma melhor fluidez na navegação e nas animações, e para os desenvolvedores, pois permite fácil acesso a funcionalidades internas do dispositivo como sistema de localização e câmeras (GU; XU; ZHENG, 2017). Por outro lado, a complexidade e o custo do desenvolvimento de uma aplicação crescem com o número de plataformas suportadas. Neste cenário, compartilhar código-fonte é praticamente impossível, dada a natureza completamente

¹ <<https://www.bemyeyes.com/>>

² <<https://ribon-landing.herokuapp.com/>>

distinta dos sistemas operacionais existentes no mercado (BIØRN-HANSEN; MAJCHRZAK; GRØNLI, 2017).

Para contornar os aspectos negativos do desenvolvimento de aplicativos nativos, surgiu o conceito de desenvolvimento híbrido: um modelo que permite o uso de tecnologias web, como HTML, CSS e Javascript para o codificação de interfaces e lógicas de negócio. Assim, através de *frameworks* especializados, responsáveis por apresentar a "página web" por meio de uma *webview* e pela comunicação entre o código desenvolvido com tecnologia web e o código nativo de cada sistema operacional, é possível desenvolver apenas um código-fonte e utilizá-lo para gerar aplicativos para diversas plataformas (BJORN-HANSEN; GROENLI; GHINEA, 2018).

Outra abordagem que tem ganhado destaque é a de produzir código nativo através de um código baseado em Javascript. Seguindo as regras determinadas pelo *framework* escolhido, é possível produzir um único código base e, com o auxílio de um compilador especializado, traduzir tal código para aquele compreendido por cada plataforma. Neste quesito, o *framework* React Native, desenvolvido pela equipe de código aberto do Facebook, tem ganhado destaque nos últimos anos. Apesar de ainda estar em sua versão beta, o *framework* já faz parte do código-fonte dos aplicativos mais utilizados do mundo, como Uber, Airbnb, Facebook e Instagram (KAKAR, 2017). Através do React Native é possível construir aplicativos móveis que são indistinguíveis daqueles escritos em Objective-C (plataforma iOS) ou Java (plataforma Android) (FACEBOOK, 2018).

2.2.5 Páginas de Captura

Página de captura, ou *landing page*, é uma página web criada especificamente com o propósito de receber o tráfego originado por ações de marketing; um meio de redirecionar os possíveis consumidores de diversas mídias para um único lugar.

Existem basicamente dois tipos de *landing page*: páginas para gerar potenciais consumidores (*leads*) a partir do registro de informações como e-mail ou telefone, e páginas projetadas para uma nova ação, por exemplo, uma compra ou o download de um app (GARDNER, 2018).

Para Dutra (2017) e Dutra (2018), o projeto de uma página de captura que atinja seu objetivo deve obedecer a alguns princípios, como:

- Simplicidade visual;
- Fácil navegação;
- Começar com uma boa imagem ou vídeo;
- Uma paleta de cores apropriada para o tipo de produto ofertado;
- Ser responsiva, proporcionando uma navegação agradável tanto em telas maiores, como a de um computador, quanto em telas menores, como a de um *smartphone*;

- Ter títulos e cabeçalhos atrativos, pois a persuasão do visitante é fator crucial para o sucesso da página;
- Chamadas para ação (*Call to Actions*) fortes, geralmente por meio de botões que se sobressaem ao conteúdo da tela.

3

Materiais e Métodos

Como apresentado no Capítulo anterior, o primeiro passo para o desenvolvimento de um software que cumpra o prazo e o orçamento estabelecidos é definir o processo de desenvolvimento que a equipe de tecnologia da informação seguirá. As metodologias ágeis ganharam bastante destaque nos últimos anos e tem se mostrado muito eficientes no gerenciamento diário da produção de software em diversas empresas ao redor do mundo (SUTHERLAND, 2014). Entregar software funcionando com frequência, adequar-se a mudanças mesmo ao fim do processo e manter contínua atenção à excelência técnica e bom design (BECK et al., 2001) alteraram o modo de produção da indústria de tecnologia e por isso foi o caminho escolhido para o desenvolvimento do LADDRES.

Como parte das atividades de gerenciamento de projeto, foram definidas micro entregas, com um intervalo não superior a três semanas. Todo o progresso de desenvolvimento foi registrado no aplicativo Trello, utilizando a metodologia Scrum, para divisão dos ciclos de entrega, e a técnica de Kanban para o acompanhamento da evolução, tanto do *sprint*, como da visão global do projeto, como mostra a Figura 1.

O versionamento do código-fonte do LADDRES foi realizado com auxílio da ferramenta Git e disponibilizado de forma aberta e livre de direitos autorais por meio do Github¹. Vale destacar a escolha por construir um projeto de código 100% aberto. Como o objetivo do LADDRES é apresentar um produto detalhado, transparente e imparcial, ter o código aberto permite que qualquer pessoa audite o funcionamento da aplicação e conheça a origem de todas as informações exibidas pelo app. Além disso, abrir o código é uma forma de retribuição para a comunidade de desenvolvedores, uma vez que todo o projeto foi desenvolvido a partir de ferramentas abertas, e de retribuição para a universidade pública que financiou indiretamente o projeto, permitindo que ele possa ser utilizado em aulas da própria instituição ou mantido por outros desenvolvedores no futuro. Além do código-fonte aberto, a maior parte dos pontos de

¹ <<https://github.com/laddres>>

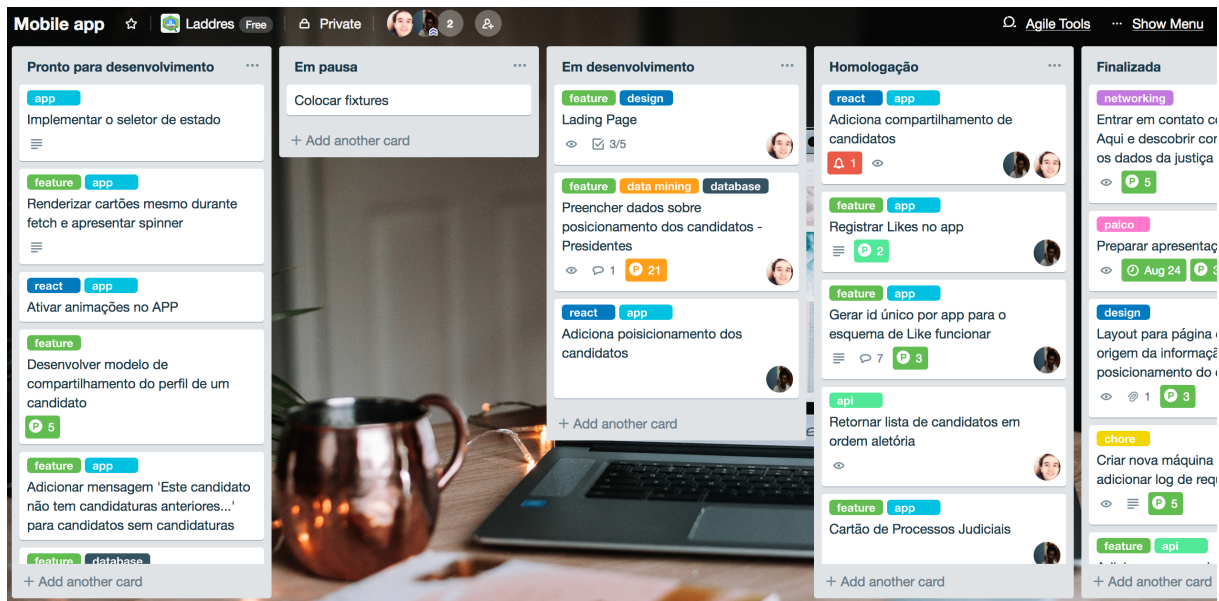


Figura 1 – Captura de tela do andamento do projeto no aplicativo Trello

acesso da API está documentada e disponível para consulta pública, possibilitando a construção de novas formas de visualização dos dados gerados durante o processo, aumentando, de forma indireta, os resultados alcançados.

Após definir a metodologia a ser aplicada e as ferramentas de controle de código e gerenciamento de projeto, o fluxo de trabalho, que incluiu as etapas de ETL (extração, transformação e carga) dos dados e a geração de conhecimento a partir deles, foi dividido em 5 etapas:

1. Obtenção dos dados, abertos ou proprietários, a partir de diversas fontes e em diferentes formatos;
2. Tratamento/transformação dos dados e geração dos *datasets*;
3. Modelagem e povoamento do banco de dados;
4. Criação de uma nova rota na API para que os dados tratados e modelados possam ser acessados pela aplicação final;
5. Desenvolvimento de módulos/componentes para o aplicativo móvel.

Para seguir o fluxo de trabalho descrito acima e demonstrado na Figura 2, de maneira organizada e escalável, o código-fonte deste trabalho foi dividido em quatro repositórios diferentes:

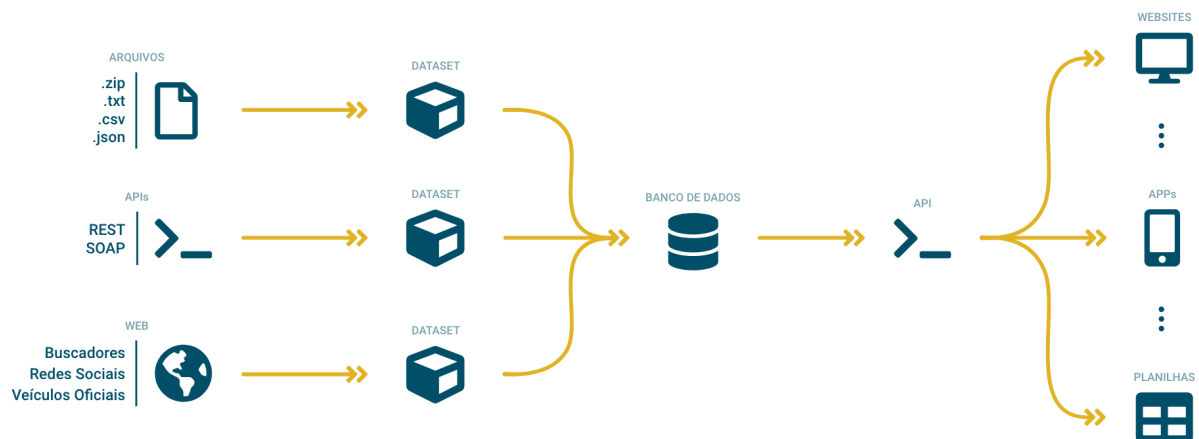


Figura 2 – Fluxo de desenvolvimento do projeto.

- Laboratório² - Ambiente de trabalho que permite experimentação e descoberta de novas possibilidades. A partir dos experimentos realizados com sucesso, são gerados datasets e as tabelas modeladas para tal funcionalidade são alimentadas;
- API³ - API pública para consulta dos dados armazenados no banco de dados do projeto. Responsável, principalmente, por fornecer os dados que serão exibidos no aplicativo móvel;
- Aplicativo⁴ - Produto final do projeto. Responsável por armazenar a interface de interação entre o LADDRES e o público em geral, disponibilizando o currículo dos candidatos às eleições de 2018 de maneira simples e direta;
- Landing Page⁵ - Página de captura e aquisição de usuários para o aplicativo, utilizada para divulgação do projeto perante os eleitores brasileiros.

Com todo processo definido, o próximo passo foi definir quais dados deveriam ser buscados e explorados.

3.1 Definição do Escopo

Quem não sabe o que busca, não identifica o que acha (KANT, 17-).

Com o principal objetivo do projeto em mente, levar informação transparente, imparcial e de qualidade para as mãos dos eleitores brasileiros, foi preciso definir quais seriam as informações

² <<https://github.com/laddres/lab>>

³ <<https://github.com/laddres/api>>

⁴ <<https://github.com/laddres/mobile-app>>

⁵ <<https://github.com/laddres/landing-page>>

mais relevantes para tais eleitores e quais delas estariam disponíveis por meio de dados abertos governamentais. Os tópicos que norteariam o “currículo LADDRES” foram então definidos baseado no desejo da população em saber sobre:

1. Honestidade: mais de 70% dos brasileiros não votariam em um candidato mencionado na Operação Lava Jato, mesmo que ainda não tenha sido condenado pela justiça (KADANUS, 2017);
2. Coerência entre discurso e ação: a maioria dos brasileiros não acredita em promessas de campanha (IBOPE, 2018);
3. Origem do candidato (CONGRESSO EM FOCO, 2018);
4. Histórico político: brasileiros valorizam experiência na política (IBOPE, 2018);
5. Posicionamento: o que o candidato pensa sobre saúde, educação, previdência, transporte e segurança (FORTUNA; RITA, 2018).

3.2 Prototipação do App

Protótipo é a tangibilização de uma ideia, a passagem do abstrato para o físico de forma a representar a realidade e propiciar validações (VIANNA, 2012). Para o desenvolvimento dos protótipos não-funcionais, a ferramenta escolhida foi o Adobe XD⁶, uma ferramenta gratuita produzida por uma das maiores empresas de *software* para *design* do mundo e que apresenta o ferramental básico necessário para desenhar e prototipar sites e aplicativos.

3.3 Obtenção e Tratamento de Dados

Por se tratar da linguagem com maior adoção no contexto da ciência de dados no mundo inteiro (MATOS, 2015), ser gratuita, simples de ser utilizada e possuir diversas bibliotecas especializadas em trabalhar com grande volume de dados como o Pandas e o Numpy, Python - versão 3 - foi a linguagem de programação escolhida para a obtenção e o tratamento dos dados neste trabalho. Além desses pontos de destaque, os *notebooks* do projeto Jupyter (PROJECT JUPYTER, 2018) são excelentes para que o desenvolvedor possa intercalar texto escrito - com opção para *markdown* - e código em *python*, possibilitando, assim, que os experimentos sejam documentados durante o processo de desenvolvimento.

⁶ <<https://www.adobe.com/br/products/xd.html>>

3.4 Banco de Dados

Por se tratar de um sistema com poucas inserções, inúmeras consultas e recheado de relações entre as entidades, o banco de dados relacional foi o modelo escolhido. Neste sentido, a escolha pelo MySQL foi uma escolha natural, dada sua natureza livre, sua larga adoção e uma comunidade de desenvolvedores grande e engajada ([STACKOVERFLOW, 2018](#)).

Para diagramação e estruturação do banco de dados de modo visual, a ferramenta escolhida foi o MySQL Workbench.

3.5 API

Para fazer a integração em tempo real entre as informações armazenadas no banco de dados e as diversas aplicações que podem ser construídas a partir desses dados, em especial o app para dispositivos móveis, é necessário desenvolver um *web service*. Levando em consideração a simplicidade de configuração e sua alta performance, para o projeto do LADDRES, a estrutura escolhida foi:

- Serviço na nuvem do Digital Ocean⁷, com sistema operacional Ubuntu 18.04;
- Nginx, como servidor HTTP;
- Linguagem de programação Javascript, através do Node.js, para implementação da API;
- Express.js como framework web;
- JSON Web Tokens, para autenticação de rotas protegidas.

3.6 Aplicativo para *Smartphones*

O primeiro desafio para o desenvolvimento de aplicações para dispositivos móveis é conseguir contemplar os dois sistemas operacionais de maior popularidade do mercado: Android e iOS. Em empresas com equipes bem estruturadas e com quantidade suficiente de pessoas, o desenvolvimento nativo é quase uma regra. Já para equipes pequenas, como a deste projeto, manter duas bases de código é absolutamente inviável.

Com o crescimento de ferramentas para desenvolvimento híbrido (um código base, compilado para n plataformas), em especial a do React Native, *framework* que já faz parte dos aplicativos mais utilizados no mundo ([FACEBOOK, 2018](#)), aliado à familiaridade com a linguagem JavaScript (a mesma utilizada no desenvolvimento da API do projeto), a escolha por construir um app híbrido com React Native foi bastante natural.

⁷ <<https://www.digitalocean.com/>>

4

Desenvolvimento

Após análise de viabilidade e custo de aquisição dos dados referentes aos tópicos definidos durante o levantamento do escopo no Capítulo 3, o primeiro passo foi desenvolver os primeiros protótipos não-funcionais do app.

4.1 Prototipação do App

Para a tela inicial, a proposta foi listar todos os candidatos a todos os cargos disponíveis nas eleições de 2018, com seus respectivos partidos e números, agrupados por cargo, como mostra a Figura 3. Ao clicar na foto do candidato, o usuário é redirecionado para página de detalhes.

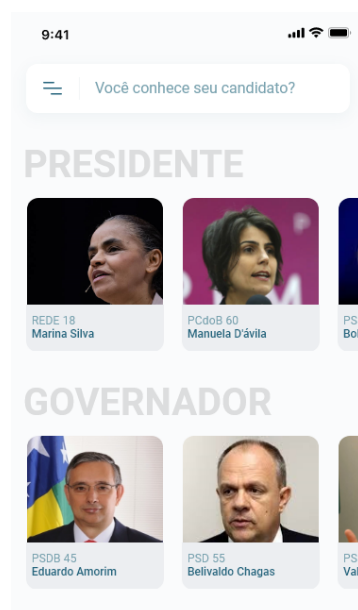


Figura 3 – Lista com todos os candidatos, agrupados por cargo.

A página de detalhes foi projetada de maneira a tentar englobar todos os tópicos citados acima em um só lugar. Para isso, foi utilizado o conceito de “cartões”, uma maneira simples e prática de separar verticalmente vários conceitos diferentes numa mesma tela da aplicação (FLIP, 2017).

O primeiro cartão da tela de detalhes do candidato é o cartão de apresentação de perfil (3º tópico do currículo). Nesta seção são mostrados dados pessoais e eleitorais como: foto do candidato, nome que aparecerá na urna, nome completo, idade, grau de escolaridade, cidade e estado de origem, número e cargo ao qual concorre, além de funcionalidades de interação com o usuário como as opções de curtir e compartilhar o perfil. O resultado pode ser visto na Figura 4.

O segundo cartão mostra um resumo da vida pública do candidato. Nesta seção são encontradas informações como: partidos aos quais o candidato foi filiado, número de processos nos quais é réu, número de candidaturas e mandatos que já exerceu, além de um resumo da atuação como parlamentar, indicando o número de proposições e de projetos apresentados caso ele tenha exercido alguma função no poder legislativo. O segundo cartão é demonstrado na Figura 5 e aborda o 4º tópico do currículo LADDRES.

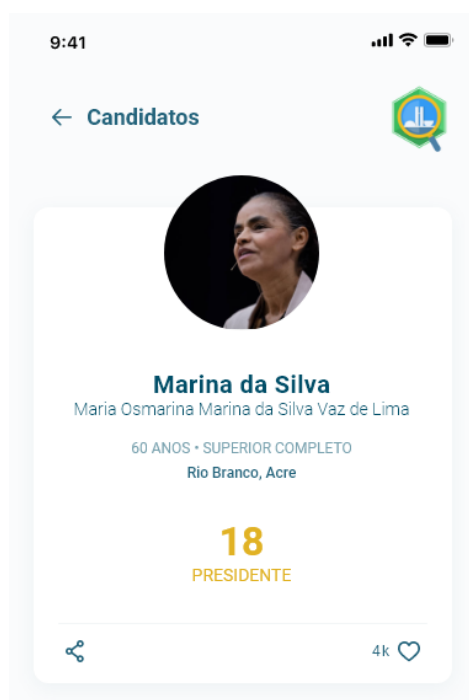


Figura 4 – Cartão com o perfil do candidato.

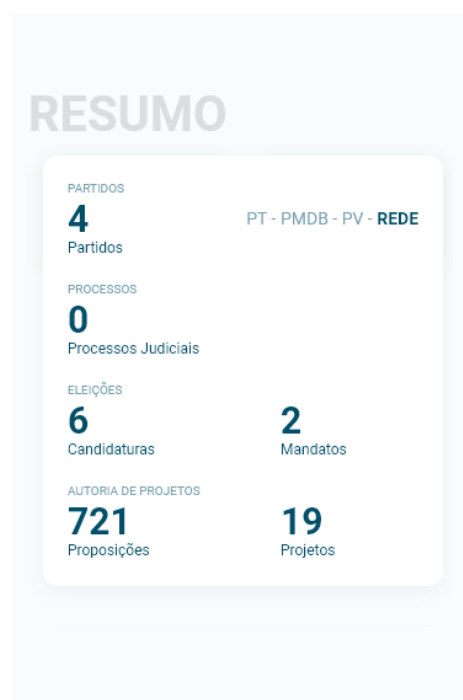


Figura 5 – Resumo da vida pública do candidato.

O terceiro cartão aborda o posicionamento do candidato - 5º tópico - sobre questões que ganharam relevância na mídia brasileira durante os últimos meses. Cada assunto pode receber uma das cinco classificações disponíveis. As classificações são “votou a favor”, “favorável”, “indefinido”, “contrária” e “votou contra”, como mostra a Figura 6. O objetivo aqui é demonstrar o quão coerente o candidato tem sido em suas votações e declarações - tópico 2. Para manter

o compromisso de transparência que o projeto tem com o eleitor, todas as classificações são justificadas por fontes seguras e que podem ser acessadas pelo app, como mostra a Figura 7. O resultado de todos estados do cartão pode ser visto nas Figuras 6, 7 e 8.



Figura 6 – Possíveis classificações para o posicionamento do candidato.

Figura 7 – Descrição do tema, posicionamento referenciado e fontes de informação.

Figura 8 – Posicionamento do candidato sobre temas que ganharam relevância.

O quarto cartão traz a lista de processos judiciais aos quais o candidato responde - 1º tópico do currículo. Além da descrição do processo, a lista conta com número e tipo da ação, tribunal responsável por seu julgamento e link para fonte da informação, como mostra a Figura 9. Caso não haja nenhum processo, o usuário encontrará a mensagem mostrada na Figura 10.

A quinta e última seção cobre os tópicos 2 e 4 do currículo, trazendo detalhes de todas as candidaturas que o candidato participou, incluindo coligações e partidos (Figura 11). Caso o candidato tenha sido eleito em alguma dessas candidaturas, o aplicativo traz detalhes de sua atuação (Figura 12), com informações como as proposições e os projetos apresentados durante o mandato, além de link para as fontes que originaram tais dados.

Definido o que deve ser procurado, o próximo passo foi definir como procurá-lo.

4.2 Obtenção e Tratamento de Dados

Por se tratar de uma aplicação para o contexto eleitoral, o melhor primeiro passo é analisar o conteúdo dos dados abertos disponibilizados pelo Tribunal Superior Eleitoral.

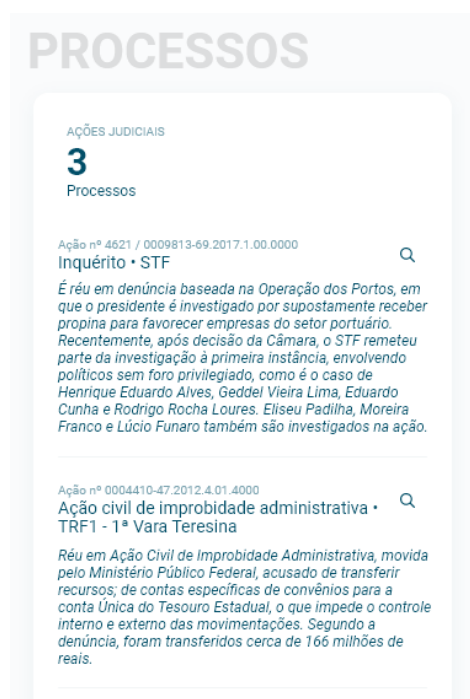


Figura 9 – Lista de processos nos quais o candidato é réu.



Figura 10 – Candidato não responde a nenhum processo judicial.



Figura 11 – Detalhes de candidaturas anteriores.



Figura 12 – Detalhes de atuação em mandatos anteriores.

4.2.1 Tribunal Superior Eleitoral

O repositório de dados eleitorais do TSE é uma compilação de informações brutas das eleições ocorridas no Brasil a partir de 1945. Ele foi desenvolvido como parte do movimento de

abertura e transparência da justiça eleitoral e é voltado para pesquisadores, imprensa e pessoas interessadas em analisar os dados de eleitorado, candidaturas, resultados e prestação de contas (TRIBUNAL SUPERIOR ELEITORAL, 2018).

Os arquivos fornecidos pelo tribunal estão no formato TXT e seu conteúdo formatado com o modelo CSV. Apesar de conter dados de mais de 70 anos de república, os dados anteriores a 2003 estão incompletos e podem apresentar falhas e inconsistências. Segundo [Tribunal Superior Eleitoral \(2018\)](#), “está sendo realizada uma revisão nas fontes de dados e, conforme os trabalhos forem concluídos, os arquivos serão substituídos”, porém, até o encerramento deste trabalho, nenhum resultado concreto neste sentido foi obtido.

O projeto enfrenta aqui sua primeira limitação e, para manter o compromisso com a transparência e a qualidade das informações, apenas as candidaturas posteriores a 2002 foram analisadas, num total de oito eleições regulares (quatro municipais e quatro federais/estaduais), além de 195 eleições extraordinárias em municípios e estados que sofreram com problemas e/ou fraudes eleitorais.

Os arquivos de texto disponibilizados pelo TSE estão agrupados por estado (cada arquivo .txt se refere a um estado brasileiro) e, junto com um arquivo de LEIAME, estão compactados em um arquivo .zip, referente ao ano da eleição. Estes arquivos podem ser obtidos em: <http://www.tse.jus.br/eleicoes/estatisticas/repositorio-de-dados-eleitorais-1/repositorio-de-dados-eleitorais>.

Para obtermos todos os arquivos de forma automatizada, utilizamos o script (disponível em https://github.com/Laddres/lab/blob/master/journal/utis/tse_download_datasets.py) que lança mão do fato de que o nome dos arquivos é padronizado (consulta_cand_ano_eleicao.zip) e itera sobre uma lista de anos geradas automaticamente a partir do ano corrente, como mostra o Código 1.

Após o download e a descompactação do arquivo .zip, é possível observar que nem todos os arquivos .txt contém uma linha de cabeçalho (*header*) embutida. Assim, o próximo passo é tratar os cabeçalhos de cada conjunto de dados. Para isso, seguimos a documentação indicada no arquivo LEIAME.pdf de cada *dataset* obtido e montamos quatro cabeçalhos diferentes:

- Arquivos de 2004 a 2010;
- Arquivos de 2012;
- Arquivos de 2014 a 2016;
- Arquivos de 2018.

O arquivo com um método para seleção dos cabeçalhos citados acima pode ser encontrado em https://github.com/Laddres/lab/blob/master/journal/utis/tse_data_headers.py

Código 1 – Script para *download* dos *datasets* do TSE

```
1 import pandas as pd
2 import sys
3 import os
4 import urllib
5 import zipfile
6
7 from utils.download import download
8
9 # Diretório temporário
10 # python3 tse_download_datasets.py ../temp
11 arg1 = sys.argv[1]
12 DIRETORIO_TEMPORARIO = os.path.join(arg1)
13
14 # Nome e endereço dos arquivos do diretório do TSE
15 PREFIXO_ARQUIVO_CANDIDATURAS = 'consulta_cand_'
16 URL_CANDIDATURAS = 'http://agencia.tse.jus.br/estatistica/sead/odsele/consulta_cand/'
17
18 # Setup de configuração para o arquivo de saída (nome do dataset)
19 HOJE = pd.datetime.today().date()
20 anos_eleitorais = [str(year) for year in (range(2004, HOJE.year - 1, 2))]
21
22 for ano in anos_eleitorais:
23     nome_arquivo = '{}{}.zip'.format(PREFIXO_ARQUIVO_CANDIDATURAS, ano)
24     url_arquivo = URL_CANDIDATURAS + nome_arquivo
25     arquivo_saida = os.path.join(DIRETORIO_TEMPORARIO, nome_arquivo)
26
27     urllib.request.urlretrieve(url_arquivo, arquivo_saida)
28
29 for ano in anos_eleitorais:
30     arquivo = PREFIXO_ARQUIVO_CANDIDATURAS + ano + '.zip'
31     path = os.path.join(DIRETORIO_TEMPORARIO, arquivo)
32
33     zip_ref = zipfile.ZipFile(path, 'r')
34     zip_ref.extractall(DIRETORIO_TEMPORARIO)
35     zip_ref.close()
```

A partir daí, um *dataset* - de dados brutos originais - com todas as candidaturas de todos os agentes públicos que concorreram a pelo menos uma eleição a partir de 2004 foi gerado, representando um arquivo final de 760 MB com 1.837.259 linhas e 46 colunas.

Levando em consideração a lista de tópicos do currículo LADDRES e fazendo uma análise detalhada nos dados contidos no *dataset* original, definiu-se, então, quais colunas deveriam ser preservadas e tratadas e quais delas poderiam ser descartadas. A partir daí, *datasets* menores foram gerados, separados por tipo de informação. A lista a seguir mostra os subconjuntos de

dados gerados a partir do conjunto original disponibilizado pelo TSE:

- Cidades: código, nome e estado;
- Estados: sigla e nome;
- Nacionalidades: código e descrição;
- Graus de Instrução: código e descrição;
- Ocupações: código e descrição;
- Eleições: ano e descrição;
- Partidos: número, sigla e nome;
- Cargos: código e nome;
- Situação da Candidatura: código e descrição;
- Resultados da Candidatura: código e descrição;
- Candidatos: nome, data de nascimento, CPF, título eleitoral, e-mail, cidade e estado de origem, nacionalidade, grau de instrução e ocupação;
- Candidaturas: candidato, eleição, cidade, estado, número do candidato, nome do candidato na urna, partido do candidato, coligação, cargo do candidato, despesa máxima de campanha, situação da candidatura e resultado da candidatura.

Com os subconjuntos gerados, partimos para o tratamento de suas informações. O objetivo desta etapa é padronizar os dados válidos, remover valores duplicados, tratar valores inválidos de acordo com o tipo de dado de cada coluna e salvar os *datasets* limpos, organizados e separados por tipo de informação.

Os principais problemas encontrados no *dataset* do TSE foram:

- Diferentes tipos de valor nulo. Exemplos: 'NaN', '#NULO#', '#NE#' e valores negativos para colunas de código (que deveriam ter valor maior que 0);
- Inexistência de uma ou mais colunas, resultando em uma linha com número de colunas menor que as 46 colunas originais, o que acaba quebrando o padrão CSV da linha inteira;
- Formatos de data sem um padrão único. Além de encontrar datas com dígitos a mais ou a menos (datas inválidas), foram encontrados diferentes modelos para uma mesma representação. Exemplos:
 - No arquivo de 2004 o formato é “01061949”;

- Nos arquivos de 2006 e a partir dos arquivos de 2012 o formato encontrado é “08/04/1945”;
- Nos anos de 2008 e 2010 o formato foi “29-DEC-69”.
- Algumas cidades de nascimento são inválidas/não existem;
- Presença de caracteres inválidos em algumas strings.

Para contornar essa situação, foi criado um conjunto de *parsers* que trata os valores válidos e padroniza os valores inválidos. Este conjunto de métodos pode ser encontrado em https://github.com/Laddres/lab/blob/master/journal/utis/tse_parsers.py

Após passar por todo esse processo, os datasets estão salvos e prontos para a etapa de povoamento do banco de dados, descrita na seção 4.3 deste documento.

4.2.2 Câmara de Deputados

O portal de Dados Abertos da Câmara pode ser utilizado tanto para acompanhar a tramitação e as votações de projetos e pareceres como para verificar a atuação e os gastos dos deputados eleitos (CÂMARA DOS DEPUTADOS, 2018).

As consultas podem ser feitas por meio de requisições HTTP a uma API RESTful ou através de arquivos disponibilizados em formato XML, JSON, CSV, XLSX (Microsoft Excel) e ODS (Open Office). A maior vantagem das consultas à API é contar com a oportunidade de trabalhar com dados abertos conectados, onde uma informação se conecta a uma cadeia de outras informações, permitindo que um *script* automatizado vasculhe todas as nuances do *dataset*; a desvantagem é que as conexões são limitadas (neste caso, em 100 resultados por requisição), um problema para quem trabalha com grande volume de dados. No momento em que a análise feita por este projeto foi realizada (entre julho e agosto de 2018), a opção de consulta aos arquivos estava muito instável e os dados inconsistentes. Assim, a mineração foi realizada via consultas à API.

Os principais pontos de acesso da API são:

- Legislaturas: períodos de trabalhos legislativos, iniciados no dia da posse dos parlamentares após uma eleição e encerrados na véspera da posse dos deputados seguintes;
- Deputados: ponto principal do *dataset*, são os agentes da câmara e atuam como autores de proposições, membros de órgãos etc. Dependendo da quantidade de votos recebidos, um deputado pode ser titular ou suplente. No Dados Abertos, cada deputado que exerceu mandato, mesmo que por apenas um dia, recebe um número identificador exclusivo;
- Partidos: organizações políticas às quais os deputados são filiados e estão oficialmente registradas no Tribunal Superior Eleitoral brasileiro;

- Órgãos: os trabalhos da Câmara são exercidos pelos deputados em diversos órgãos parlamentares como comissões, CPIs, mesa diretora, procuradoria, conselhos etc;
- Proposições: além das proposições mais conhecidas como Projetos de Lei e Propostas de Emendas à Constituição (PECs), há também outros tipos de documentos legislativos como pareceres, requerimentos, relatórios, medidas provisórias entre tantas outras. A lista completa pode ser obtida em <<https://dadosabertos.camara.leg.br/api/v2/referencias/tiposProposicao>>;
- Votações: proposições de diversos tipos são submetidas a votações e algumas dessas votações são nominais, com o registro dos votos de cada deputado.

Analisando as possibilidades ofertadas pelos dados abertos da câmara e baseado nos tópicos definidos para o currículo LADDRES, foram buscados dados que colaborassem para preencher os tópicos de “coerência entre discurso e ação”, ao capturar suas proposições e votações no plenário, e “histórico político”, ao capturar as ações enquanto parlamentar.

O primeiro passo foi tentar entender como a câmara de deputados funciona. No total são 40 tipos de órgão, 420 tipos de proposição e 213 tipos de tramitação. Além disso, cada órgão pode ter 19 status (situação em que se encontra) diferentes e cada proposição até 80 deles.

A partir disso, foram gerados *datasets* para:

- Listar todas as legislaturas: a partir desse dado é possível separar a atuação parlamentar por mandato e conectá-la às respectivas candidaturas (geradas pelos dados do TSE);
- Listar todos os órgãos: a fim de auxiliar a identificação dos órgãos que cada deputado integrou durante o mandato;
- Listar todas as proposições;
- Listar todos os deputados;

Vale ressaltar que os *endpoints* de órgãos, proposições e deputados desta API retornam dados simplificados, como mostra o Código 2

Para obter um *dataset* robusto, com o maior nível de detalhe possível, foi necessário realizar consultas individuais a cada um dos resultados obtidos nas listas descritas acima. O nível detalhado para um deputado é mostrado no Código 3. Para delimitar o escopo deste projeto, a obtenção de detalhes de proposições (que totalizavam 398.386 resultados) foi restringida aos tipos PEC, PL, PLP, PFC, PDC e MPV.

Código 2 – Retorno simplificado de um Deputado Federal

```
1 {
2   "id": 178957,
3   "uri": "https://dadosabertos.camara.leg.br/api/v2/deputados/178957",
4   "nome": "ABEL MESQUITA JR.",
5   "siglaPartido": "DEM",
6   "uriPartido": "https://dadosabertos.camara.leg.br/api/v2/partidos/36769",
7   "siglaUf": "RR",
8   "idLegislatura": 55,
9   "urlFoto": "http://www.camara.leg.br/internet/deputado/bandep/178957.jpg"
10 }
```

Código 3 – Retorno detalhado para um Deputado Federal

```
1 {
2   "dados": {
3     "id": 178957,
4     "uri": "https://dadosabertos.camara.leg.br/api/v2/deputados/178957",
5     "nomeCivil": "ABEL SALVADOR MESQUITA JUNIOR",
6     "ultimoStatus": {
7       "id": 178957,
8       "uri": "https://dadosabertos.camara.leg.br/api/v2/deputados/178957",
9       "nome": "ABEL MESQUITA JR.",
10      "siglaPartido": "DEM",
11      "uriPartido": "https://dadosabertos.camara.leg.br/api/v2/partidos/36769",
12      "siglaUf": "RR",
13      "idLegislatura": 55,
14      "urlFoto": "http://www.camara.leg.br/internet/deputado/bandep/178957.jpg",
15      "data": "2015-02-01",
16      "nomeEleitoral": "ABEL MESQUITA JR.",
17      "gabinete": {
18        ...
19      },
20      "situacao": "Exercício",
21      "condicaoEleitoral": "Titular",
22      "descricaoStatus": null
23    },
24    ...
25  },
26  "links": [
27    {
28      "rel": "self",
29      "href": "https://dadosabertos.camara.leg.br/api/v2/deputados/178957"
30    }
31  ]
32 }
```

Um conjunto de funções foi desenvolvido para auxiliar as tarefas de obtenção e transformação de dados da API em dataframes do Pandas. Estes métodos estão disponíveis em https://github.com/Laddres/lab/blob/master/journal/utis/camara_obter_dados.py

Ao fim do processo, os *datasets* foram capazes de sintetizar as seguintes informações:

- Deputados: nome, nome completo, data de nascimento, data de falecimento, sexo, partido, estado e município de origem, estado pelo qual foi eleito, endereço do gabinete parlamentar e *website*;
- Mandatos: deputado e legislatura;
- Tipos de proposição: código e nome;
- Proposições: identificador, autor, tipo de proposição, número, ano e ementa.

O maior complicador desta etapa foi o fato de, pelo volume de requisições feitos aos servidores da câmara, o tempo de resposta e a rejeição da mesma terem se tornado algo frequente. Se por um lado houve facilidade em inter-relacionar os dados disponíveis, por outro, houve uma certa dificuldade em consegui-los em um menor espaço de tempo.

4.2.3 Processos Judiciais

O poder judiciário brasileiro é bem complexo. São diversos tribunais estaduais e federais, quatro instâncias e três ramos especializados da justiça federal da união (FEDERAL, 2018), como mostra a Figura 13. Apesar de estar passando por um processo de modernização e de abertura de dados nos últimos anos, ainda não é possível obter informações públicas de maneira centralizada e automatizada no país.



Figura 13 – Organograma do Poder Judiciário

Fonte: <https://estranhosidade.files.wordpress.com/2014/10/organograma-judiciario-inv-m.jpg>

Para conseguir incorporar o primeiro tópico do currículo (Honestidade) ao aplicativo do LADDRES, foi preciso buscar fontes secundárias (que não são as originais) e confiáveis para tentar uma parceria e construir uma ferramenta mais completa.

Uma das iniciativas da sociedade civil que tem demonstrado destaque na união de dados dos diversos tribunais espalhados pelo país é o Vigie Aqui, “o jeito mais fácil de descobrir quem são os políticos enrolados com a justiça” ([RECLAME AQUI, 2018b](#)). Idealizado pelo instituto Reclame Aqui, um dos maiores sites do Brasil - com mais de 600.000 acessos diários ([RECLAME AQUI, 2018a](#)), o projeto conseguiu levantar dados judiciais de presidentes, ex-presidentes, senadores, deputados federais e governadores, formando o *dataset* mais completo da esfera político-judicial brasileira.

Após alguns contatos, o LADDRES e o Vigie Aqui firmaram uma parceria e o *dataset* de processos judiciais foi disponibilizado via arquivo .json, parcialmente demonstrado pelo Código 4.

Com o arquivo em mãos, o processo de desenvolvimento foi similar aos citados acima:

- Padronização dos dados válidos:
 - CPF e Título Eleitoral: remover máscaras, adicionar zeros à esquerda e remover caracteres extras;
 - Descrição dos processos: remover caracteres especiais e espaços em branco no início e no fim das strings.
- Remoção de valores duplicados;
- Tratamento de valores inválidos de acordo com o tipo de dado de cada coluna, no geral, transformando em valor nulo;
- Armazenamento dos *datasets* limpos e organizados.

Após a organização de dois conjuntos de dados, um para pessoas e outro para processos, foi feita uma conexão entre dados do Vigie Aqui e o *dataset* de candidatos gerado anteriormente, através dos campos:

- CPF do candidato;
- Título eleitoral, caso o campo CPF seja nulo;
- E-mail institucional da câmara, para o caso de o candidato ser um deputado federal e os dados não apresentarem nenhuma das duas informações anteriores.

Código 4 – Parte do arquivo disponibilizado pelo projeto Vigie Aqui

```
1 {
2   "id": 1777,
3   "names": [
4     "Lídice da Mata e Souza",
5     "Lídice da Mata"
6   ],
7   "display_name": "Lídice da Mata",
8   "photo": "https://s3.amazonaws.com/vigieaqui-static-files/pessoa/850.jpg",
9   "cpf": "146.720.495-15",
10  "electoral_card": "0.394.329.905-82",
11  "political_party": "PSB",
12  "contact": "lidice.mata@senadora.leg.br",
13  "updated_at": "2018-08-31T17:47:35-03:00",
14  "mandates": [
15    {
16      "office": "Senador",
17      "state": "BA",
18      "type": "Estadual",
19      "year_start": "2011",
20      "year_end": "2018",
21      "candidate": false,
22      "elected": true
23    },
24    ...
25  ],
26  "lawsuits": [
27    {
28      "id": 7216,
29      "name": "Inquérito 4396/2017",
30      "type": "Inquérito",
31      "number": "4396/2017",
32      "department": "STF",
33      "link": "http://www.stf.jus.br/...?...&tipoJulgamento=M",
34      "description": {
35        "pt-br": "É investigada no inquérito [...] com o apelido de ‘Feia’",
36        "en": "[...]"
37      },
38      "updated_at": "2018-05-03T17:58:32-03:00"
39    }
40  ],
41  "purple": true
42 }
```

Todo o processo descrito acima está disponível nos *notebooks* do Jupyter dos dias 2018-09-01 a

2018-09-03 em <<https://github.com/Laddres/lab/tree/master/journal>>.

Assim, os *datasets* estão prontos para a etapa de povoamento do banco de dados, descrita na seção 4.3 deste documento.

4.2.4 Dados Não Estruturados

Por fim, para abordar o último tópico do currículo, saber o que o candidato pensa/como age em relação aos mais variados temas, foi necessário um levantamento sobre quais os assuntos que ganharam mais destaque nos últimos anos. Os tópicos escolhidos foram:

- Congelamento dos Gastos em Serviços Públicos;
- Reforma Trabalhista;
- Terceirização de Todas as Atividades;
- Reforma da Previdência;
- Taxação de Grandes Fortunas e Heranças;
- Reforma do Ensino Médio;
- Cobrança por Cursos de Especialização em Universidades Públicas;
- Cotas em Universidades Públicas;
- Isenção Fiscal a Petroleiras Estrangeiras;
- Escola sem Partido;
- Fim do Regime de Partilha;
- Privatização da Eletrobrás;
- Porte de Armas;
- Redução da Maioridade Penal;
- Intervenção Federal (Militar) no Rio de Janeiro;
- Descriminalização do Aborto;
- Descriminalização das Drogas.

Apesar de os dados do poder legislativo informarem como alguns dos deputados e senadores votaram em alguns dos tópicos listados acima, nem todos os assuntos foram levados a

votação em plenário. Além disso, apenas uma pequena parcela dos candidatos às eleições de 2018 cumpriram mandato nas últimas legislaturas.

Não há nenhuma estrutura de dados abertos que trate desses elementos e a própria natureza do discurso ambíguo dos políticos brasileiros dificulta o trabalho de algoritmos de aprendizagem conseguirem tirar conclusões precisas.

Assim, não foi possível desenvolver um método automatizado para se obter o posicionamento de cada candidato, individualmente, em jornais, revistas, redes sociais, programas de televisão ou qualquer outra fonte de dados não estruturados. Como cumprir esta tarefa manualmente, para todos os candidatos, demandaria uma equipe com centenas de pessoas ou um prazo de vários meses (ou anos), o escopo da funcionalidade de posicionamento foi delimitado para candidatos que tiveram mandatos na última legislatura e/ou candidatos à presidência do país.

Construíram-se, então, dois *datasets*: um com a lista de temas, incluindo descrição e links relacionados; e outro com a lista de posicionamento dos candidatos.

4.3 Banco de Dados

Dado o caráter incremental do modelo de desenvolvimento deste projeto, a modelagem do banco de dados também foi feita aos poucos, com novas versões a cada nova fonte de dados integrada. Com o auxílio da ferramenta MySQL Workbench, foi montado o diagrama demonstrado pela Figura 14 cujo objetivo é mostrar a evolução do banco de dados.

A área 1 da Figura 14 representa as tabelas construídas a partir dos *datasets* gerados na seção 4.2.1 deste documento. Os dados do TSE são responsáveis por trazer informações sobre a origem e o histórico político do candidato, além de controlar todo o fluxo relativo às eleições, como partidos, cargos, situações e resultados eleitorais.

As tabelas da área 2 da Figura 14 armazenam os dados obtidos através da API da câmara de deputados. Através da conexão entre a tabela Deputado e a tabela Candidato é possível identificar as proposições de sua autoria, órgãos dos quais foi membro e seu posicionamento em cada votação.

Na parte superior central, área 3 da Figura 14, encontram-se as tabelas criadas a partir do conjunto de dados disponibilizado pelo Vigie Aqui. A tabela *politico_vigieaqui* faz a conexão entre o módulo de processos judiciais e o restante da aplicação através de uma chave-estrangeira para a tabela Candidato.

A parte inferior central, área 4 da Figura 14, é a responsável pelos temas coletados para a construção do módulo de posicionamento. Além dos temas e das posições dos candidatos, nela está presente tabelas auxiliares que armazenam links de apoio e a categorização dos temas e das posições.

Por último, no canto superior direito, área 5 da Figura 14, estão duas tabelas que permitem

que o app armazene as curtidas únicas de cada candidato: a tabela *usuario* registra o dispositivo que está acessando este módulo enquanto a tabela *like* armazena as curtidas (um par de dispositivo-candidato).

Assim, o script mostrado no Código 5 foi criado para migrar os dados dos *datasets* para suas respectivas tabelas no banco de dados. O código também está disponível em <https://github.com/Laddres/lab/blob/master/journal/utils/dataframe_to_database.py>

Código 5 – Migração de dados de um Pandas Dataframe para o banco de dados MySQL

```
1 import pandas as pd
2 import numpy as np
3 import mysql.connector as mysql
4
5 BD_CONFIG = {
6     'user': 'root',
7     'password': '',
8     'host': '127.0.0.1',
9     'database': 'laddres'
10 }
11
12 def dataframe_to_database(dataframe, tabela, colunas_tabela, colunas_dataframe):
13     cnx = mysql.connect(**BD_CONFIG)
14     cursor = cnx.cursor()
15
16     for row in dataframe.itertuples():
17         columns = ', '.join(colunas_tabela)
18         placeholders = ', '.join(np.full((1, len(colunas_tabela)), '%s')[0])
19
20         sql = 'INSERT INTO {} ({} ) VALUES ({} )'.format(tabela, columns, placeholders)
21         values = []
22         for column in colunas_dataframe:
23             values.append(getattr(row, column))
24
25         try:
26             cursor.execute(sql, tuple(values))
27             cnx.commit()
28         except:
29             print(row)
30             cursor.close()
31             cnx.close()
32             raise
33
34     cursor.close()
35     cnx.close()
```

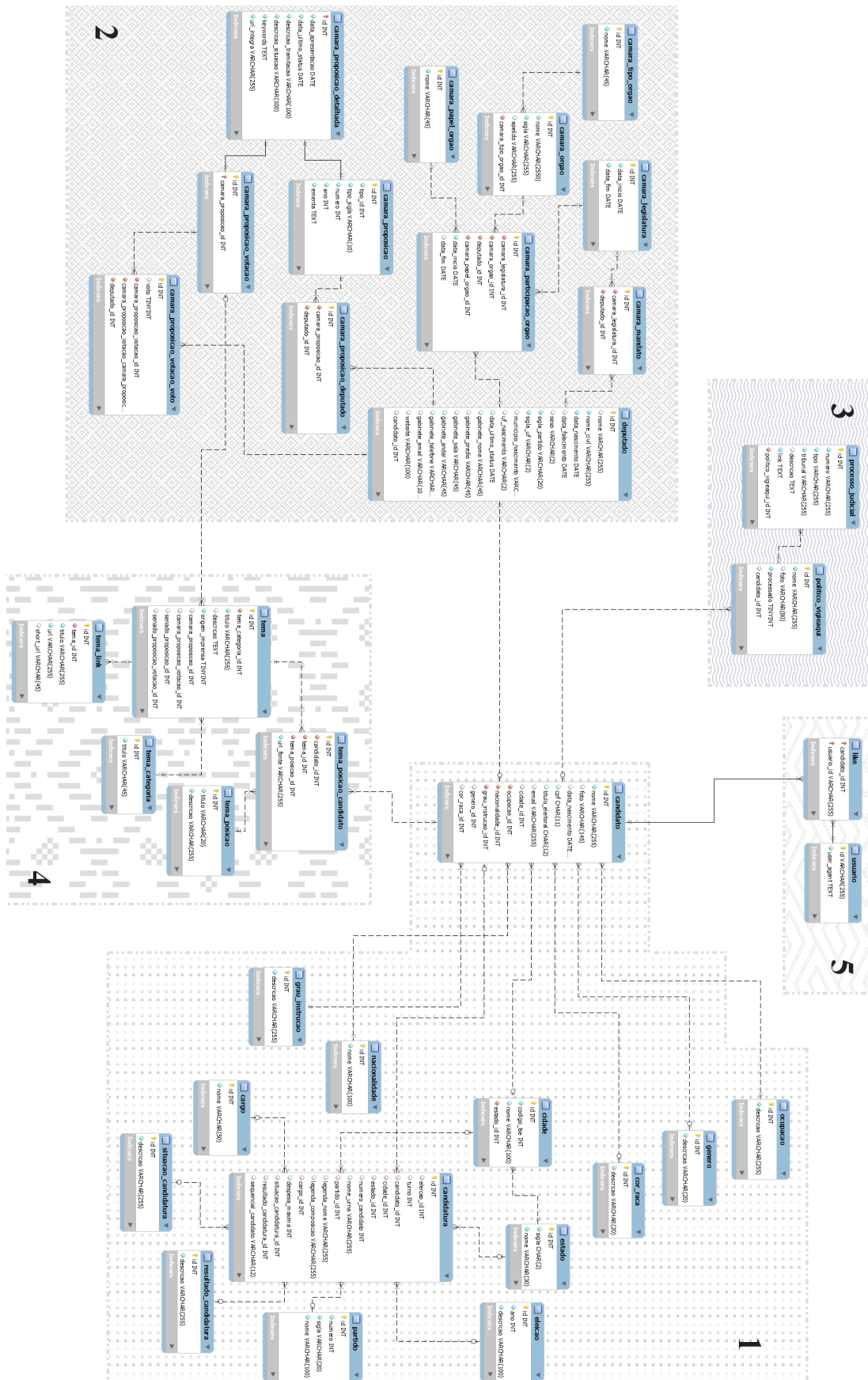


Figura 14 – Estrutura do Banco de Dados do Projeto

Vale ressaltar a importância da tabela de *candidato* para o projeto. Cada novo módulo se conecta a ela, permitindo encontrar, a partir de seu id, todas as informações importantes para o usuário final.

Outro ponto importante é perceber que a maioria dos campos aceita valor nulo, inclusive as chaves-estrangeiras. Apesar de ser uma má prática de projeto em banco de dados, foi necessário configurá-las assim, já que muitos dos dados coletados estavam incompletos, inconsistentes ou simplesmente não se conectariam com as tabelas relacionadas e exigiriam uma estrutura muito complexa para contornar essa quebra no padrão (o custo disso não compensaria os benefícios propiciados). Apesar disso, os dados anuláveis não interferem no funcionamento do processo e foram armazenados numa tentativa de recuperação futura através da integração com outros *datasets*. O código completo para *setup* das tabelas mostradas na Figura 14 pode ser encontrado em <https://github.com/Laddres/lab/blob/master/src/database/database-setup.sql>.

Por se tratar de uma base de dados muito grande, com mais de quatro milhões de entradas, em que a maioria das consultas requer uma quantidade considerável de *joins* entre as tabelas, o tempo de resposta para algumas consultas estava demorando mais do que o aceitável para um aplicativo móvel, numa média superior a 20 segundos. Para contornar este problema, foram criados *scripts* para geração de “*hot tables*”, tabelas enxutas, otimizadas e que agregam várias informações num único lugar. Ao invés de realizar uma consulta com 8 *inner joins*, por exemplo, a construção de *hot tables* específicas para cada eleição possibilitou, utilizando-se as mesmas configurações de hardware e software, reduzir, de 40 segundos para uma média de 600 milissegundos, o tempo de uma consulta a um candidato através de seu nome. O diagrama com as *hot tables* é mostrado na Figura 15. O Código 6 mostra um exemplo de *script* SQL utilizado para geração de uma *hot table*. O código para geração de todas as *hot tables* pode ser encontrado em <https://github.com/laddres/lab/tree/master/journal/utlis/sqls>.

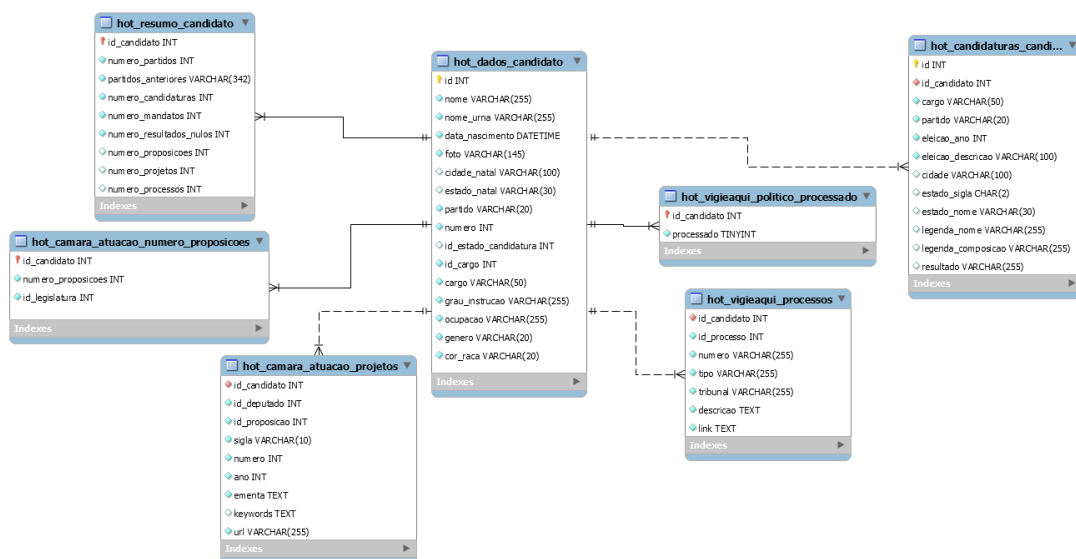


Figura 15 – Estrutura das *hot tables* geradas.

Código 6 – Exemplo de algoritmo para geração de uma *hot table*.

```
1 CREATE OR REPLACE VIEW
2   view_dados_candidato
3 AS
4   SELECT
5     candidatura.candidato_id as id,
6     candidato.nome,
7     candidatura.nome_urna,
8     candidato.data_nascimento,
9     candidato.foto,
10    cidade.nome as cidade_natal,
11    estado.nome as estado_natal,
12    partido.sigla as partido,
13    candidatura.numero_candidato as numero,
14    candidatura.estado_id as id_estado_candidatura,
15    candidatura.cargo_id as id_cargo,
16    cargo.nome as cargo,
17    grau_instrucao.descricao as grau_instrucao,
18    ocupacao.descricao as ocupacao
19 FROM
20   candidatura
21   LEFT JOIN partido on candidatura.partido_id = partido.id
22   LEFT JOIN cargo on candidatura.cargo_id = cargo.id
23   LEFT JOIN candidato on candidatura.candidato_id = candidato.id
24   LEFT JOIN cidade on candidato.cidade_id = cidade.id
25   LEFT JOIN estado on cidade.estado_id = estado.id
26   LEFT JOIN grau_instrucao on candidato.grau_instrucao_id = grau_instrucao.id
27   LEFT JOIN ocupacao on candidato.ocupacao_id = ocupacao.id
28 WHERE
29   (candidatura.eleicao_id = 202 OR candidatura.eleicao_id = 203) AND
30   (
31     candidatura.situacao_candidatura_id IS NULL OR
32     candidatura.situacao_candidatura_id IN (2, 4, 8, 12, 16, 17, 18, 19)
33   );
34
35 CREATE TABLE hot_dados_candidato AS SELECT * FROM view_dados_candidato;
36
37 CREATE INDEX id_index ON hot_dados_candidato (id);
38 CREATE INDEX nome_index ON hot_dados_candidato (nome);
39 CREATE INDEX nome_urna_index ON hot_dados_candidato (nome_urna);
40
41 DROP VIEW view_dados_candidato;
```

4.4 API

Baseado no padrão REST, a API do LADDRES é composta por diversas rotas que identificam as entidades do sistema. Para atender as necessidades levantadas na etapa de prototipação, as rotas foram divididas em:

- `/candidatos` - *endpoint* que retorna uma lista com os dados básicos de todos os candidatos elegíveis na atual eleição, divididos por cargo (ex.: presidente, governador etc). Além disso, este resultado pode ser filtrado por meio de *query strings*, por exemplo, o *endpoint* `/candidatos?nome=João` retornará todos os candidatos que possuam nome João;
- `/candidatos/:id` - *endpoint* responsável por retornar o perfil detalhado do candidato através do seu identificador único (id);
- `/candidatos/:id/resumo` - retorna o resumo da vida pública de um determinado candidato. Traz informações como os números de partidos, candidaturas, proposições, mandatos e processos judiciais.
- `/candidatos/:id/posicionamentos` - retorna a posição de (alguns) candidatos em relação a temas que estão em alta nestas eleições;
- `/candidatos/:id/processos` - retorna a lista de processos judiciais em que um determinado candidato é réu;
- `/candidatos/:id/candidaturas` - retorna os detalhes de todas as candidaturas de um determinado candidato;
- `/candidatos/:id/mandatos` - retorna o número total de proposições apresentadas por um determinado candidato e detalhes sobre os projetos (do tipo PL, PEC, PFC, PLP, PDC) de sua autoria.

Além dessas rotas públicas, há também a configuração de algumas rotas privadas. O motivo da restrição de algumas rotas é a necessidade de se manter o controle sobre algumas funcionalidades do sistema para, por exemplo, evitar fraudes no sistema de curtidas (impedindo voto de pessoas/sistemas não autorizados) ou evitar acesso irrestrito a dados de terceiros (como é o caso dos dados do Vigie Aqui). Para contornar este problema, foi utilizado o sistema de *tokens* que funciona com o seguinte fluxo:

1. Ao ser aberta pelo usuário, a aplicação requisita um *token* de acesso informando uma chave secreta, que somente ela e o servidor conhecem;
2. O servidor autentica a chave enviada e, através da biblioteca `jsonwebtoken`, cria um token criptografado, exclusivo, para o dispositivo que o solicitou;

3. A aplicação salva o *token* recebido e o utiliza sempre que acessar uma rota protegida.

A documentação detalhada e todo o código-fonte da API do LADDRES podem ser encontrados em <https://github.com/laddres/api>.

Vale destacar que para alternar entre as configurações de desenvolvimento local e de produção, o LADDRES utiliza o padrão de projeto em que um arquivo `.env` é criado para cada um dos ambientes de programação e nele são guardadas as variáveis que podem mudar de acordo com o ambiente utilizado (WIGGINS, 2017). Neste arquivo ficam guardadas as configurações do MySQL - com endereço do servidor, nome de usuário, senha e nome da tabela - e as chaves de verificação utilizadas para proteção das rotas por meio do JSON Web Token.

Após o desenvolvimento do código e após configurar o sistema operacional na máquina virtual do serviço de nuvem¹, instalar e configurar o Nginx² - a configuração de produção do LADDRES é mostrada no Código 7 -, instalar um certificado de segurança para utilização de conexões seguras por meio do protocolo HTTPS³ e fazer o setup de um ambiente de produção para Node.js⁴, a API está pronta para ser lançada em produção e ser consumida pelas aplicações.

Código 7 – Configuração do Nginx para servir a API do LADDRES

```
1 server {
2     listen 80;
3     listen [::]:80;
4
5     access_log /var/log/nginx/laddres-api-access.log timed;
6     error_log /var/log/nginx/laddres-api-error.log;
7
8     root /var/www/caueira.com.br/laddres/api/build;
9     index index.html;
10
11     server_name api.caueira.com.br;
12
13     location / {
14         proxy_pass http://localhost:3000;
15         proxy_http_version 1.1;
16         proxy_set_header Upgrade $http_upgrade;
17         proxy_set_header Connection 'upgrade';
18         proxy_set_header Host $host;
19         proxy_cache_bypass $http_upgrade;
20     }
21 }
```

¹ <https://www.digitalocean.com/community/tutorials/configuracao-inicial-de-servidor-com-ubuntu-18-04-pt>

² <https://www.digitalocean.com/community/tutorials/how-to-install-nginx-on-ubuntu-18-04>

³ <https://www.digitalocean.com/community/tutorials/how-to-secure-nginx-with-let-s-encrypt-on-ubuntu-18-04>

⁴ <https://www.digitalocean.com/community/tutorials/how-to-set-up-a-node-js-application-for-production-on-ubuntu-18-04>

4.5 Aplicativo para *Smartphones*

Apesar de não haver um padrão de projeto bem definido para as aplicações construídas com o React Native, algumas iniciativas *open source* têm ganhado destaque. Uma das principais é a estrutura “Andross”⁵ proposta pela Ignite CLI, da Infine Red, empresa que desenvolve apps para gigantes como Microsoft e Universidade da Califórnia (INFINITE RED, 2018). A estrutura inclui, entre outros *frameworks*, os seguintes destaques:

- React Native: *framework* base para o desenvolvimento de aplicativos híbridos, principalmente para Android e iOS;
- React Navigation: *framework* para controle de rotas e navegação dentro do app;
- Redux: *framework* para controlar os estados da aplicação;
- Redux Sagas: *framework* para controlar o fluxo de requisições a recursos externos ao app;
- Storybook: *framework* que permite “contar histórias” sobre os componentes da aplicação. Com ele é possível expor todas as aparências possíveis de um componente.

E a arquitetura do aplicativo é definida por:

- Containers: na maioria dos casos é uma tela completa, mas pode ser parte de uma tela, dependendo da necessidade do desenvolvedor. São componentes que controlam o estado da aplicação;
- Navigation: arquivos responsáveis pela navegação no app, definindo rotas e separando interesses;
- Components: são os componentes criados com React, desenvolvidos de uma maneira que maximize sua reutilização no app;
- Storybook: arquivos com as histórias dos componentes;
- Themes: arquivos de tema, relacionados à parte visual do aplicativo. Exemplos clássicos são arquivos para armazenar variáveis de cor, métricas e fontes;
- Config: arquivos de configuração do sistema, como variáveis de ambiente e ferramentas de *debug*;
- Fixtures: arquivos que “imitam” as respostas da API para acelerar o processo em ambientes de desenvolvimento;

⁵ <<https://github.com/infinitered/ignite-ir-boilerplate-andross>>

- Redux, Sagas: arquivos que controlam fluxo de informações na aplicação através da alteração de estados e do acesso a recursos fora do app, como, por exemplo, requisições a um API;
- Services: arquivos com serviços e utilitários para o app;
- Libs: espaço para módulos que poderão ser transformados em bibliotecas no futuro;
- Images: arquivos de imagem (.png) utilizados pelo app;
- Transforms: métodos que auxiliam o tratamento de dados recebidos via API, transformando-os para o modelo desejado pelo aplicativo;
- Tests: arquivos de teste do app, incluindo testes unitários e *snapshots*.

Com a arquitetura da aplicação bem definida e com base nos protótipos expostos no início deste capítulo, o aplicativo móvel do LADDRES foi desenvolvido e seu código-fonte pode ser encontrado em <<https://github.com/Laddres/mobile-app/tree/develop>>.

Em resumo, foram criados dois *containers*, um para cada tela: tela principal e tela de currículo. A tela principal foi dividida em vários componentes, os principais foram: barra de busca, seções (para os cargos) e o cartão contendo a foto, o nome e o partido do candidato. Já a tela de resumo foi dividida em um número maior de componentes; para cada seção do currículo há basicamente dois componentes, um para a seção como um todo e um para o cartão com o conteúdo detalhado. Além desses, alguns componentes auxiliares também foram desenvolvidos, como o componente ImageButton (para criar botões a partir de ícones) e o componente Separator (para separar conteúdos dentro dos cartões).

Após todo desenvolvimento e sequência de testes em grupos fechados, o aplicativo foi lançado na Play Store, para dispositivos Android, e na Apple Store, para dispositivos iOS.

Com o app disponibilizados nas lojas, a última etapa é construir a página de captura para divulgação do LADDRES.

4.6 Página de Captura

A construção da página de captura (*landing page*) como ferramenta de marketing e divulgação do LADDRES buscou obedecer os princípios expostos por Dutra (2017) e Dutra (2018) para obter uma *landing page* de sucesso. Com simplicidade visual, conteúdo dividido em blocos, navegação de acesso rápido e uma chamada para ação (Call to Action) muito forte, a página de captura <<https://quemsaooscandidatos.com.br/>> é o canal utilizado para que usuários do país inteiro conheçam o projeto e usufruam de seu conteúdo. Parte desta página é demonstrada na Figura 16.



Figura 16 – Página de captura do projeto

Escrita apenas com HTML, CSS e Javascript, a página foi construída em cima do modelo MobApp⁶ da Colorlib⁷ e lapidada para atender as necessidades do projeto. Seu código-fonte pode ser acessado em <https://github.com/laddres/landing-page> e completa, assim, o ciclo de desenvolvimento do projeto LADDRES, que saiu de uma ideia inicial, passou pela prototipação e validação de hipótese, pela obtenção e tratamento de dados de diversas fontes diferentes, modelagem de um banco de dados robusto, criação de APIs de alto valor agregado, até chegar na produção e divulgação do produto final, um aplicativo que leva o currículo dos candidatos até as mãos dos eleitores.

⁶ <https://colorlib.com/wp/template/mobapp/>

⁷ <https://colorlib.com>

5

Resultados e Discussões

Ao final de todo processo descrito acima, os resultados deste trabalho podem ser divididos em dois tipos: qualitativo e quantitativo. O primeiro refere-se ao objetivo final do trabalho: levar informação transparente, imparcial e de qualidade aos eleitores brasileiros. Já o segundo refere-se ao volume de informação trafegado pelo *pipeline* desenvolvido durante o percurso.

5.1 Qualitativo

Baseado nos protótipos apresentados anteriormente e considerando as limitações encontradas durante o *pipeline* de mineração de dados descrito nos capítulos anteriores, produziu-se, então, uma API pública para consulta estruturada a todas as informações obtidas e um aplicativo móvel disponibilizado para *smartphones* que possuem sistema operacional Android¹ e iOS².

5.1.1 API Pública

Além de servir como fonte para todas as requisições realizadas pelo aplicativo LADRES, a API desenvolvida durante este trabalho foi disponibilizada publicamente para que outros desenvolvedores pudessem, a partir dessas informações, gerar novos métodos de visualização dos dados. Dados como a lista e os detalhes de todos os candidatos, lista de candidaturas anteriores e atuações em mandatos de cada um dos políticos puderam ser utilizados, a partir desta API, como base para o desenvolvimento de sites, planilhas e relatórios relacionados ao processo eleitoral de 2018.

Um exemplo de resultado retornado por uma consulta à API que tem por objetivo listar todos os candidatos a todos os cargos destas eleições é mostrado no Código 8. A documentação completa do projeto da API pode ser encontrada em <<https://github.com/laddres/api>>.

¹ <<https://play.google.com/store/apps/details?id=com.laddres>>

² <<https://itunes.apple.com/br/app/laddres/id1436180000>>

Código 8 – Retorno de uma consulta à API de Candidatos

```
1 {  
2   "presidente": [  
3     {  
4       "id": 402793,  
5       "nome": "Eymael",  
6       "foto": "http://divulgacandcontas.tse.jus.br/.../foto_1534343806232.jpg",  
7       "numero": 27,  
8       "partido": "DC"  
9     },  
10    ...  
11  ],  
12  "governador": [ ... ],  
13  "senador": [ ... ],  
14  "deputadoFederal": [ ... ],  
15  "deputadoEstadual": [ ... ]  
16 }
```

5.1.2 Aplicativo Móvel

A primeira interação do LADDRES com os usuários se dá através da indicação da localidade dos mesmos. Ao abrir o aplicativo pela primeira vez, o usuário é solicitado a permitir acesso a sua geolocalização, Figura 17. Esta identificação é importante para determinar os candidatos que serão exibidos para os cargos do governo local, como governadores, senadores e deputados. Após ler a seção de dúvidas frequentes, Figura 18, o usuário terá acesso a tela principal do app. Caso ele não tenha permitido o reconhecimento automático de sua localização, um seletor de estados será exibido antes da transição para a próxima tela, como mostra a Figura 19.

A tela principal, Figura 20, traz a foto, o nome, o partido e o número de todos os candidatos a todos os cargos eletivos para o estado selecionado. Além disso, uma barra de busca é responsável por filtrar os candidatos listados através de seu nome completo ou nome registrado para a urna eleitoral. Ainda nesta tela, é possível clicar no botão de filtro e exibir a tela responsável por uma filtragem mais detalhada. A Figura 21 mostra todas as opções de filtro disponíveis. Nela, o usuário pode:

- Exibir apenas os candidatos marcados como favoritos;
- Exibir apenas candidatos que concorrem a sua primeira eleição;
- Filtrar pelos gêneros definidos pelo TSE: feminino ou masculino;
- Filtrar pelas raças/cores definidas pelo TSE: branca, preta, parda, amarela ou indígena;

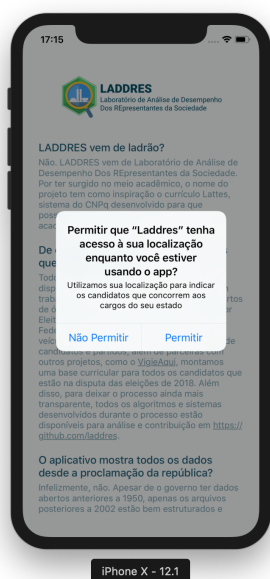


Figura 17 – Solicitação de acesso à localização do usuário.

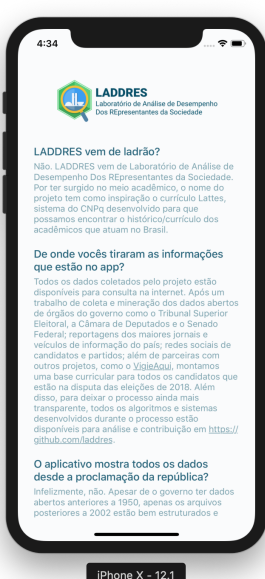


Figura 18 – Respostas a perguntas frequentes de usuários.



Figura 19 – Seletor para indicar a localização do usuário.

- Selecionar qualquer outra unidade federativa brasileira.

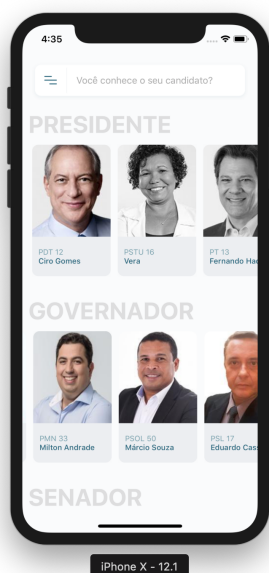


Figura 20 – Tela principal, exibe todos os candidatos desta eleição.

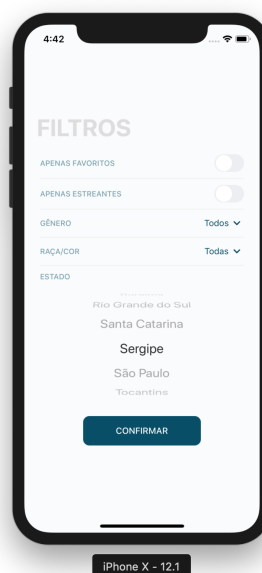


Figura 21 – Tela de seleção de filtros para exibição dos candidatos.

Ao selecionar um candidato na tela principal, o usuário é redirecionado para a página que contém seu “currículo”. Nesta tela são mostrados o perfil - Figura 22 -, o resumo da vida pública - Figura 23 -, o histórico de processos judiciais - Figura 23 e Figura 24 -, e o histórico de

candidaturas anteriores - Figura 25. Vale ressaltar que todas as informações mostradas nesta tela do aplicativo vem acompanhadas de links que redirecionam o usuário para as respectivas fontes e íntegras. Para isto, basta clicar no botão de lupa disponíveis no canto superior direito de cada processo e/ou projeto apresentado, como mostram as Figuras 24 e 25.

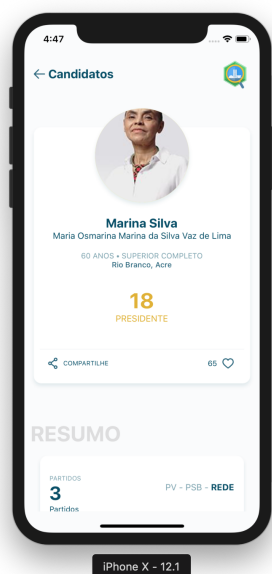


Figura 22 – Cartão com o perfil do candidato.



Figura 23 – Cartões com resumo e processos de um candidato.



Figura 24 – Cartão com os processos judiciais de um candidato.



Figura 25 – Cartão com o histórico de candidaturas de um candidato.

5.2 Quantitativo

Ao longo do desenvolvimento do projeto descrito neste trabalho, foram processados milhares de arquivos de diversos formatos, gerado pouco mais de 1 GB em *datasets* e registradas mais de quatro milhões de entradas no banco de dados.

A análise dos dados do Tribunal Superior Eleitoral, a partir de 2004, produziu um *dataset* bruto - com todas as informações originais - de 779 MB. Além disso, apenas para as eleições de 2018, foram tratados mais de 213 MB de fotos de candidatos. Ao todo foram processadas 1.866.357 candidaturas correspondentes a 1.274.037 candidatos.

Apesar de não apresentarem o mesmo volume que os dados do TSE, os dados da câmara de deputados e de processos judiciais também produziram um volume considerável de informação. Ao todo foram capturados 1.478 deputados, 53.764 participações em órgãos da câmara e 398.386 proposições, produzindo mais de 100 MB em *datasets* com dados tratados e relacionados. Já na parte dos processos judiciais, os dados foram mais modestos: ao todo foram 3.027 políticos investigados e 1.157 processos judiciais, produzindo cerca de 5 MB em *datasets*.

Além disso, durante a última semana de corrida eleitoral para o primeiro turno das eleições foram registrados pouco mais de 900 downloads em dispositivos com sistema operacional Android (Play Store) e pouco mais de 11.300 downloads na loja de aplicativos da Apple (Apple Store), totalizando mais de doze mil downloads em ambas plataformas. Ao total foram recebidas 43 avaliações com nota máxima e uma média de 4,6 pontos, de 5 possíveis. Vale destacar, também, que o aplicativo desenvolvido neste trabalho permaneceu no top 10 apps da Apple Store na categoria utilitários durante todo o período de votação do primeiro turno (dia 07 de outubro de 2018).

Outra estatística que merece destaque é o número de acessos à API desenvolvida neste trabalho. Somente durante o final de semana dos dias 5, 6 e 7 de outubro, foram registradas pouco mais de 3 milhões de requisições realizadas a partir de diversos países do mundo.

6

Considerações Finais

Este trabalho mostrou como, a partir de conjuntos de dados de diversas fontes, é possível extrair informações que podem auxiliar em uma escolha tão difícil quanto a de definir os representantes de uma nação pelos próximos quatro, ou até oito anos.

Apesar de contar com uma equipe bem reduzida, com apenas dois integrantes, foi possível passar por todo processo de construção de uma solução em tecnologia da informação: partindo apenas de uma ideia, passando pela extração, transformação e a carga de dados de diferentes fontes e formatos, pelo projeto completo de uma estrutura de banco de dados, pelo desenvolvimento de modelos de acesso às informações, até chegar a divulgação de um produto pronto e disponível para a sociedade.

Comparativamente, projetos similares ao LADDRES (como alguns citados no capítulo introdutório deste documento) contam com equipes multidisciplinares com dezenas, ou até centenas de integrantes. Conseguir atingir o objetivo proposto inicialmente, com um escopo tão grande e um time tão pequeno, mostra a eficácia do processo desenvolvido neste trabalho.

Vale ressaltar a expressividade dos resultados obtidos: mais de 12.000 *downloads* do aplicativo e mais de 3.000.000 de acessos à API pública em apenas uma semana de divulgação e lançamento do projeto. Sem nenhum investimento financeiro em ferramentas de marketing e divulgação, os números foram alcançados de maneira orgânica por uma equipe enxuta e sua rede de contatos.

Outro ponto que chama atenção é a disparidade entre o número de usuários nas plataformas Android e iOS. Enquanto na loja de aplicativos da Apple o *app* alcançou, e se manteve por mais de 24 horas, no top 10 da lista de mais baixados, encontrar o aplicativo na loja de aplicativos do Google se mostrou um problema a ser resolvido para as próximas versões.

Para a melhoria do *pipeline* de mineração desenvolvido neste projeto, o primeiro passo é separar o código de experimentação do código de produção. Apesar de ser completamente viável

reproduzir os experimentos descritos neste trabalho sem essa separação, a reorganização de todo o código facilitaria a entrada de novos colaboradores e agilizaria o processo de reprodução.

Como continuação do projeto LADDRES, o próximo passo é consumir os dados abertos do Senado Federal. Apesar de não ser tão bem estruturado quanto os dados da Câmara de Deputados, o Senado também disponibiliza informações sobre os parlamentares, os projetos e as votações; seria um modo relativamente simples de enriquecer o currículo dos candidatos, mas que, pela falta de tempo disponível, não foi possível incorporar ao escopo deste trabalho.

Além dos dados do senado, o maior desafio para as eleições de 2020 será minerar dados de prefeituras e câmaras de vereadores de todas as cidades brasileiras. Por ser um país de dimensões continentais e inúmeras particularidades regionais, desenvolver um modelo de acesso e tratamento a todos esses dados é um desafio que exigirá o apoio da comunidade organizada em times multidisciplinares. Entretanto, desenvolver um modelo capaz de minerar apenas os dados das capitais brasileiras já seria um bom pontapé inicial para tornar a ferramenta mais completa e atingir o grande objetivo inicial do projeto: levar informação de qualidade para que as escolhas sejam feitas de maneira cada vez mais consciente, fortalecendo esta jovem e frágil democracia brasileira.

Referências

- ABOUTORABI, S. H. et al. Performance evaluation of sql and mongodb databases for big e-commerce data. In: IEEE. *Computer Science and Software Engineering (CSSE), 2015 International Symposium on*. [S.l.], 2015. p. 1–7. Citado na página 20.
- AGUNE, R. M.; FILHO, A. S. G.; BOLLIGER, S. P. Governo aberto sp: disponibilização de bases de dados e informações em formato aberto. 2010. Citado na página 16.
- APOIA.SE. *Operação Serenata de Amor*. 2018. Disponível em: <<https://apoia.se/serenata>>. Acesso em: 26 set. 2018. Citado na página 12.
- BECK, K. et al. Manifesto for agile software development. 2001. Citado 2 vezes nas páginas 19 e 24.
- BIØRN-HANSEN, A.; MAJCHRZAK, T. A.; GRØNLI, T.-M. Progressive web apps: The possible web-native unifier for mobile development. In: *Proceedings of the 13th International Conference on Web Information Systems and Technologies (WEBIST)*. [S.l.: s.n.], 2017. p. 344–351. Citado na página 22.
- BIRGEN, C.; PREISIG, H.; MORUD, J. Sql vs. nosql. *Norwegian University of Science and Technology, Scholar article*, 2014. Citado na página 19.
- BJORN-HANSEN, A.; GROENLI, T.; GHINEA, G. A survey and taxonomy of core concepts and research challenges in cross-platform mobile development. 2018. Citado na página 22.
- BOUCHEZ, Y.; GATINOIS, C. *Corruption aux Jeux de Rio : le CIO suspend le comité olympique brésilien*. 2017. Disponível em: <http://www.lemonde.fr/jeux-olympiques-rio-2016/article/2017/10/06/l-etau-se-resserre-autour-du-patron-des-jeux-de-rio_5197123_4910444.html>. Acesso em: 26 abr. 2018. Citado na página 11.
- BRASIL. *Lei de Acesso à Informação*. Diário Oficial da República Federativa do Brasil, 2011. Lei Nº 12.527. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2011-2014/2011/lei/112527.htm>. Acesso em: 26 set. 2018. Citado na página 18.
- BUSSLER, C. Sql for nosql databases: Déjà vu. In: *CIDR*. [S.l.: s.n.], 2015. Citado na página 20.
- CÂMARA DOS DEPUTADOS. *Dados Abertos da Câmara dos Deputados*. 2018. Disponível em: <<https://dadosabertos.camara.leg.br/swagger/api.html>>. Acesso em: 26 set. 2018. Citado na página 36.
- CASTILLO, P. A. et al. Soap vs rest: Comparing a master-slave ga implementation. *arXiv preprint arXiv:1105.4978*, 2011. Citado na página 21.
- CHARLEAUX, J. ao P. *O que foram, afinal, as Jornadas de Junho de 2013. E no que elas deram*. 2017. Disponível em: <<https://www.nexojornal.com.br/expresso/2017/06/17/O-que-foram-afinal-as-Jornadas-de-Junho-de-2013.-E-no-que-elas-deram>>. Acesso em: 26 set. 2018. Citado na página 11.
- CHIGNARD, S. A brief history of open data. *ParisTech Review*, 03 2013. Citado na página 15.

CISNEIROS, V. C. de A.; RAMOS, H. Comparação entre servidores web apache http server e nginx. 2015. Citado na página 20.

CONGRESSO EM FOCO. *Eleitor quer candidato crente em Deus, honesto, com experiência e de origem pobre*. 2018. Disponível em: <<https://congressoemfoco.uol.com.br/especial/noticias/eleitor-quer-candidato-crente-em-deus-honesto-com-experiencia-e-de-origem-pobre/>>. Acesso em: 26 set. 2018. Citado na página 27.

CORREIA, L. C. *Impeachment Dilma. As melhores pérolas dos deputados para justificarem o voto*. 2016. Disponível em: <<https://laurochammacorreia.jusbrasil.com.br/noticias/325696683/impeachment-dilma-as-melhores-perolas-dos-deputados-para-justificarem-o-voto>>. Acesso em: 26 set. 2018. Citado na página 12.

CRAWFORD, T.; HUSSAIN, T. A comparison of server side scripting technologies. In: *Proceedings of the 15th International Conference on Software Engineering Research and Practice, Las Vegas, NV*. [S.l.: s.n.], 2017. p. 69–76. Citado na página 21.

DATA SCIENCE BRIGADE. *Operação Serenata de Amor: Inteligência Artificial para Controle Social da Administração Pública*. 2018. Disponível em: <<https://serenata.ai/>>. Acesso em: 26 set. 2018. Citado na página 12.

DB-ENGINES. *DB-Engines Ranking - popularity ranking of database management systems*. 2018. Disponível em: <<https://db-engines.com/en/ranking>>. Acesso em: 26 set. 2018. Citado na página 20.

DEMARTINI, F. *Brasil já tem mais de um smartphone ativo por habitante*. 2018. Disponível em: <<https://canaltech.com.br/produtos/brasil-ja-tem-mais-de-um-smartphone-ativo-por-habitante-112294/>>. Acesso em: 26 set. 2018. Citado na página 21.

DINIZ, V. Como conseguir dados governamentais abertos. 2010. Citado na página 17.

DUTRA, R. *Melhores Práticas de Design Para Sites “One Page”*. 2017. Disponível em: <<http://designr.com.br/melhores-praticas-de-design-para-sites-one-page/>>. Acesso em: 26 set. 2018. Citado 2 vezes nas páginas 22 e 51.

DUTRA, R. *Como Criar o Design Perfeito Para Landing Pages*. 2018. Disponível em: <<http://designr.com.br/como-criar-o-design-perfeito-para-landing-pages/>>. Acesso em: 26 set. 2018. Citado 2 vezes nas páginas 22 e 51.

ELLINGWOOD, J. *Apache vs Nginx: Practical Considerations*. [S.l.]: DigitalOcean, 2015. Citado na página 20.

FACEBOOK. *Who’s using React Native?* 2018. Disponível em: <<https://facebook.github.io/react-native/showcase.html>>. Acesso em: 26 set. 2018. Citado 2 vezes nas páginas 22 e 28.

FEDERAL, S. T. *Sistema Judiciário Brasileiro: organização e competências*. 2018. Disponível em: <<https://stf.jusbrasil.com.br/noticias/2535347/sistema-judiciario-brasileiro-organizacao-e-competencias>>. Acesso em: 26 set. 2018. Citado na página 39.

FILHO, F. d. H. B. A crise econômica de 2014/2017. *Estudos Avançados*, scielo, v. 31, p. 51 – 60, 04 2017. ISSN 0103-4014. Disponível em: <http://www.scielo.br/scielo.php?script=sci_arttext&pid=S0103-40142017000100051&nrm=iso>. Citado na página 11.

- FLIP, C. *Best practices for designing cards*. 2017. Disponível em: <<https://uxplanet.org/best-practices-for-designing-cards-a19f53cab052>>. Acesso em: 26 set. 2018. Citado na página 30.
- FOLHA DE SÃO PAULO. *Operação Lava Jato*. 2018. Disponível em: <<http://arte.folha.uol.com.br/poder/operacao-lava-jato/>>. Acesso em: 26 set. 2018. Citado na página 11.
- FORTUNA, D.; RITA, B. S. *Pesquisa do Correio revela o que o eleitor quer dos governantes*. 2018. Disponível em: <https://www.correiobraziliense.com.br/app/noticia/politica/2018/05/28/interna_politica,684026/pesquisa-do-correio-revela-o-que-o-eleitor-quer-dos-governantes.shtml>. Acesso em: 26 set. 2018. Citado na página 27.
- FRANCINO, Y. *Scrum vs. Kanban: How to combine the best of both methods*. 2018. Disponível em: <<https://techbeacon.com/scrum-vs-kanban-how-combine-best-both-methods>>. Acesso em: 26 set. 2018. Citado na página 19.
- G1. *Brasil fica em 96º lugar em ranking de 2017 dos países menos corruptos*. 2018. Disponível em: <<https://g1.globo.com/mundo/noticia/brasil-fica-em-96-lugar-entre-180-paises-no-ranking-da-corrupcao-de-2017.ghtml>>. Acesso em: 26 set. 2018. Citado na página 12.
- GARDNER, O. *The Unbounce Landing Page Course*. 2018. Disponível em: <<https://thelandingpagecourse.com/>>. Acesso em: 26 set. 2018. Citado na página 22.
- GENARI, J. O. S.; FERRARI, F. C. Times de alto desempenho no contexto das metodologias scrum e kanban. *Revista TIS*, v. 4, n. 3, 2016. Citado na página 19.
- GOMES, A. F. *Agile: Desenvolvimento de software com entregas frequentes e foco no valor de negócio*. [S.l.]: Editora Casa do Código, 2014. Citado na página 19.
- GRAY, J. et al. *Manual dos Dados Abertos: Governo*. São Paulo: Comitê Gestor da Internet no Brasil, 2011. Disponível em: <http://www.w3c.br/pub/Materiais/PublicacoesW3C/Manual_Dados_Abertos_WEB.pdf>. Citado na página 12.
- GU, Y.; XU, C.; ZHENG, M. Using react native in an android app. In: *MICS2017-Conference Proceedings of Midwest Instruction and Computing Symposium*. [S.l.: s.n.], 2017. Citado na página 21.
- IBOPE. Expectativas para as eleições 2018. *Indicadores CNI*, v. 7, n. 43, mar 2018. ISSN 2317-7012. Disponível em: <<https://static.congressoemfoco.uol.com.br/2018/03/cni-ibope.pdf>>. Acesso em: 26 set. 2018. Citado na página 27.
- INFINITE RED. *Mobile App Web Developers*. 2018. Disponível em: <<https://infinite.red/>>. Acesso em: 26 set. 2018. Citado na página 50.
- ISOTANI, S.; BITTENCOURT, I. *Dados Abertos Conectados: em Busca da Web do Conhecimento*. [S.l.: s.n.], 2015. ISBN 978-85-7522-449-6. Citado na página 18.
- JOKURA, T. *Só 5 presidentes eleitos completaram o mandato em 90 anos*. 2016. Disponível em: <<https://super.abril.com.br/historia/so-5-presidentes-eleitos-completaram-o-mandato-nos-ultimos-90-anos/>>. Acesso em: 26 set. 2018. Citado na página 11.

- KADANUS, K. *Político tradicional já era: maioria dos eleitores quer candidato novo em 2018*. 2017. Disponível em: <<https://www.gazetadopovo.com.br/politica/republica/politico-tradicional-ja-era-maioria-dos-eleitores-quer-candidato-novo-em-2018-ebie1x2ldjkw57jxin2vzwj>>. Acesso em: 26 set. 2018. Citado na página 27.
- KAKAR, K. *10 Amazing Apps that are Built using React Native*. 2017. Disponível em: <<https://insights.daffodilsw.com/blog/10-amazing-apps-that-are-built-using-react-native>>. Acesso em: 26 set. 2018. Citado na página 22.
- KANT, I. *Quem não sabe o que busca, não identifica o que acha*. 17–. Citado na página 26.
- LEITE, A. *Copa do Mundo da corrupção saqueou arenas construídas pelo Brasil*. 2017. Disponível em: <<http://esportes.estadao.com.br/noticias/futebol,copa-do-mundo-da-corrupcao-saqueou-arenas-construidas-pelo-brasil,70001781996>>. Acesso em: 26 set. 2018. Citado na página 11.
- MARTINS, J. C. C. *Técnicas para gerenciamento de projetos de software*. [S.l.]: Brasport, 2007. Citado na página 18.
- MATOS, D. *Por que Cientistas de Dados escolhem Python?* 2015. Disponível em: <<http://www.cienciaedados.com/por-que-cientistas-de-dados-escolhem-python/>>. Acesso em: 26 set. 2018. Citado na página 27.
- MIERS, F. *The 5 Main Agile Methodologies: Which One Is Best For You?* 2017. Disponível em: <<https://www.businesscomputingworld.co.uk/the-5-main-agile-methodologies-which-one-is-best-for-you/>>. Acesso em: 26 set. 2018. Citado na página 19.
- MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÃO. *Sobre a plataforma Lattes*. 2018. Disponível em: <<http://memoria.cnpq.br/web/portal-lattes/sobre-a-plataforma>>. Acesso em: 26 set. 2018. Citado na página 13.
- MUMBAIKAR, S.; PADIYA, P. et al. Web services based on soap and rest principles. *International Journal of Scientific and Research Publications*, v. 3, n. 5, p. 1–4, 2013. Citado na página 21.
- NUNES, R. D. A implantação das metodologias ágeis de desenvolvimento de software scrum e extreme programming (xp): uma alternativa para pequenas empresas do setor de tecnologia da informação. *ForScience*, v. 4, n. 2, 2017. Citado na página 19.
- OBAMA, B. *Memorandum on Transparency and Open Government*. 2009. Disponível em: <<https://obamawhitehouse.archives.gov/the-press-office/transparency-and-open-government>>. Acesso em: 26 set. 2018. Citado na página 15.
- OPEN DEFINITION. *The Open Definition*. 2018. Disponível em: <<https://opendefinition.org/>>. Acesso em: 26 set. 2018. Citado na página 15.
- OPEN GOVERNMENT DATA. *The Annotated 8 Principles of Open Government Data*. 2018. Disponível em: <<https://opengovdata.org/>>. Acesso em: 26 set. 2018. Citado na página 17.
- OPEN GOVERNMENT PARTNERSHIP. *What is the Open Government Partnership?* 2018. Disponível em: <<https://www.opengovpartnership.org/about/about-ogp>>. Acesso em: 26 set. 2018. Citado na página 18.

- OPEN KNOWLEDGE INTERNATIONAL. *Global Open Data Index*. 2016. Disponível em: <https://index.okfn.org/place/>. Acesso em: 26 set. 2018. Citado na página 12.
- OPEN KNOWLEDGE INTERNATIONAL. *What is Open?* 2018. Disponível em: <https://okfn.org/opendata/>. Acesso em: 26 set. 2018. Citado na página 16.
- PORTAL BRASILEIRO DE DADOS ABERTOS. *Concursos, Hackatons e Outros Eventos*. 2018. Disponível em: <http://dados.gov.br/concursos>. Acesso em: 26 set. 2018. Citado na página 18.
- PROJECT JUPYTER. *Project Jupyter*. 2018. Disponível em: <http://jupyter.org/>. Acesso em: 26 set. 2018. Citado na página 27.
- QEDU. *QEDu: Use dados. Transforme a educação*. 2018. Disponível em: <http://www.qedu.org.br/>. Acesso em: 26 set. 2018. Citado na página 13.
- RECLAME AQUI. *Sobre o Reclame Aqui*. 2018. Disponível em: <https://www.reclameaqui.com.br/institucional/>. Acesso em: 26 set. 2018. Citado na página 40.
- RECLAME AQUI. *Vigie Aqui - O plugin que vai mudar a política brasileira*. 2018. Disponível em: <http://www.vigieaqui.com.br/>. Acesso em: 26 set. 2018. Citado na página 40.
- RIBEIRO, C. J. S.; ALMEIDA, R. F. d. Dados abertos governamentais (open government data): instrumento para exercício de cidadania pela sociedade. 2013. Citado 2 vezes nas páginas 15 e 16.
- SARDINHA, E. *Cerca de 150 deputados investigados participam da votação do impeachment*. 2016. Disponível em: <http://congressoemfoco.uol.com.br/noticias/cerca-de-150-deputados-investigados-participam-da-votacao-do-impeachment/>. Acesso em: 26 set. 2018. Citado na página 12.
- STACKOVERFLOW. *Newest MySql Questions*. 2018. Disponível em: <https://stackoverflow.com/questions/tagged/mysql>. Acesso em: 26 set. 2018. Citado na página 28.
- SUTHERLAND, J. *SCRUM: A arte de fazer o dobro de trabalho na metade do tempo*. [S.l.]: Leya, 2014. Citado na página 24.
- TANNER, M.; WILLINGH, U. von et al. Factors leading to the success and failure of agile projects implemented in traditionally waterfall environments. *Human Capital without Borders: Knowledge and Learning for the Quality of Life*. Portoroz, Slovenia: Make Learn, 2014. Citado na página 18.
- THE APACHE SOFTWARE FOUNDATION. *Apache HTTP Server Project*. 2018. Disponível em: <http://httpd.apache.org/>. Acesso em: 26 set. 2018. Citado na página 20.
- TRIBUNAL SUPERIOR ELEITORAL. *Repositório de Dados Eleitorais*. 2018. Disponível em: <http://www.tse.jus.br/eleicoes/estatisticas/repositorio-de-dados-eleitorais-1/repositorio-de-dados-eleitorais>. Acesso em: 26 set. 2018. Citado na página 33.
- VAUPEL, S. et al. Model-driven development of mobile applications for android and ios supporting role-based app variability. *Software & Systems Modeling*, Springer, v. 17, n. 1, p. 35–63, 2018. Citado na página 21.

VAZ, J. C.; RIBEIRO, M. M.; MATHEUS, R. Dados governamentais abertos e seus impactos sobre os conceitos e práticas de transparência no brasil. *Cadernos ppg-au/ufba*, v. 9, n. 1, 2010. Disponível em: <<https://portalseer.ufba.br/index.php/ppgau/article/download/5111/3700>>. Citado na página 17.

VIANNA, M. *Design thinking: inovação em negócios*. [S.l.]: Design Thinking, 2012. Citado na página 27.

WIGGINS, A. *Store config in the environment*. 2017. Disponível em: <<https://12factor.net/config>>. Acesso em: 26 set. 2018. Citado na página 49.

YOUTUBE. *Vote no Tiririca, pior que tá, não fica!* 2010. Disponível em: <<https://www.youtube.com/watch?v=QQ39IWNaqAQ>>. Acesso em: 26 set. 2018. Citado na página 11.